



دانشگاه آزاد اسلامی واحد نجف آباد

گروه مهندسی کامپیوتر

عنوان پروژه:

طراحی نرم افزار برگزاری آزمون

استاد راهنما: حسین هادی پور صنعتی

ارائه دهنده: محمد نوید حیدری هرستانی

آذرماه - ۱۴۰۲



دانشگاه آزاد اسلامی  
واحد نجف آباد

چکیده:

این پروژه شامل یک اپلیکیشن ویندوز فرم است که از دیتابیس برای مدیریت اطلاعات استفاده می کند. این اپلیکیشن مربوط به مدیریت آزمون و نمرات دانش آموزان می باشد. در زیر یک چکیده به زبان فارسی از این پروژه آمده است:

توضیح پروژه:

پروژه حاضر یک اپلیکیشن ویندوز فرم است که برای مدیریت آزمون ها و نمرات دانش آموزان ساخته شده است. این برنامه از دیتابیس برای ذخیره اطلاعات استفاده می کند و دارای قابلیت های متعددی از جمله ثبت نام داوطلبان، ارتباط با دیتابیس، مدیریت آزمون ها و نمرات است.

اجزای کلیدی:

\*\*\*فرم جدید دانش آموز\*\*\*

- این فرم اطلاعات داوطلبان را ثبت می کند، شامل نام، نام خانوادگی، کد ملی و عکس.

\*\*\*فرم ایجاد نمره\*\*\*

- این بخش نمرات دانش آموزان را در آزمون ها ثبت می کند. در نظر گرفته شده است که امکان کسر امتیاز در صورت پاسخ های اشتباه وجود داشته باشد.

\*\*\*فرم مدیریت آزمون\*\*\*

- این بخش برای مدیریت آزمون ها، شامل ایجاد آزمون جدید و مشاهده آمار و نتایج استفاده می شود.

\*\*\*فرم ویرایش سوال\*\*\*

- در این بخش می توانید سوالات مربوط به آزمون ها را ویرایش کرده و تغییرات را در دیتابیس ذخیره کنید.

ابزارها و تکنولوژی ها استفاده شده:

- \*\*\*زبان برنامه نویسی C#\*\*\*

- برای توسعه برنامه از زبان برنامه نویسی C# استفاده شده است.

- \*\*\*دات نت (DotNet)\*\*\* - از فریم ورک دات نت برای توسعه برنامه استفاده شده است.

- \*\*دیتابیس SQL Server:\*\*

- جهت ذخیره و مدیریت اطلاعات داوطلبان، آزمون‌ها و نمرات از دیتابیس SQL Server استفاده شده است.

قابلیت‌ها و کدهای کلیدی:

- \*\*ثبت نام داوطلبان:\*\*

- برنامه قادر است اطلاعات داوطلبان را در دیتابیس ثبت نماید و در صورت وجود داوطلب با کد ملی تکراری، اعلام می‌کند.

- \*\*محاسبه نمره با کسر امتیاز:\*\*

- در صورت تنظیم مشخصات مربوط به کسر امتیاز برای پاسخ‌های اشتباه، نمرات با توجه به این قانون محاسبه می‌شوند.

- \*\*ویرایش سوالات آزمون:\*\*

- قابلیت تغییر و ویرایش سوالات مربوط به آزمون‌ها در دیتابیس وجود دارد.

این پروژه یک مدیریت کامل برای آزمون‌ها و نمرات دانش‌آموزان فراهم می‌کند و از دیتابیس برای ذخیره اطلاعات استفاده می‌کند.

## فهرست:

### فصل اول :

مقدمه ۱-۱.....	۶
----------------	---

### فصل دوم : تحلیل سیستم آزمون گیر

تجزیه و تحلیل ۱-۲.....	۱۱
------------------------	----

• نمودار های uml.....	۱۴
-----------------------	----

• طراحی پایگاه داده.....	۳۲
--------------------------	----

• پیاده سازی گرافیکی.....	۴۴
---------------------------	----

پیاده سازی نرم افزاری ۲-۲.....	۵۱
--------------------------------	----

• پیاده سازی بخش مدیریت داوطلبان	
----------------------------------	--

• پیاده سازی بخش مدیریت آزمون	
-------------------------------	--

• پیاده سازی بخش برگزاری آزمون	
--------------------------------	--

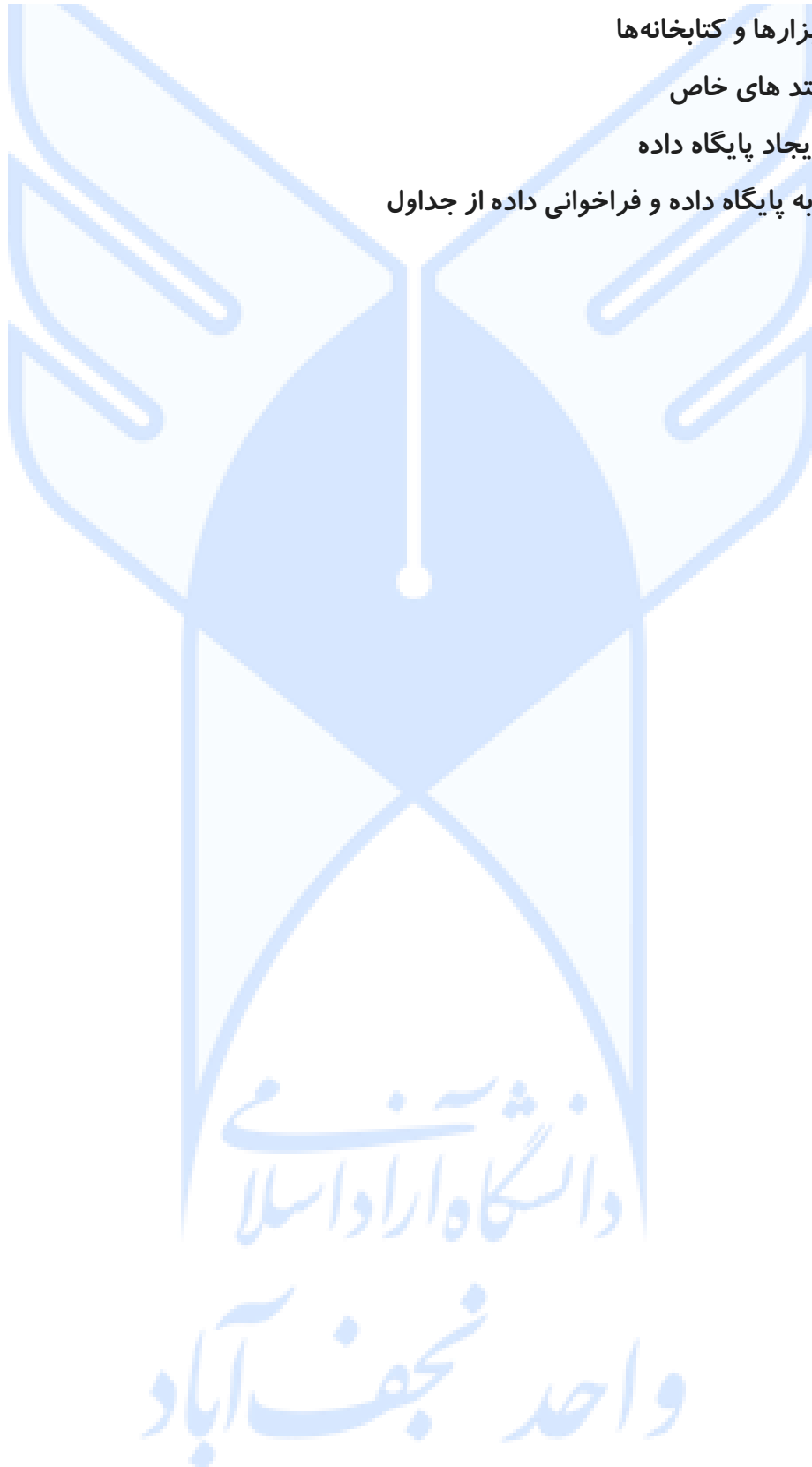
• پیاده سازی بخش نتایج	
------------------------	--

• پیاده سازی بخش گزارش گیری	
-----------------------------	--

• پیاده سازی بخش امکانات نرم افزار	
------------------------------------	--

فصل سوم: تکنولوژی ها و محیط های کاربردی پروژه.....	۱۴
--	----

- ۱-۳ محیط برنامه نویسی و پیاده سازی
- ۲-۳ ابزارها و کتابخانه‌ها
- ۳-۳ متد های خاص
- ۴-۳ ایجاد پایگاه داده
- اتصال به پایگاه داده و فراخوانی داده از جداول



## فصل اول :

### مقدمه

آزمون‌گیری یکی از روش‌های مهم ارزیابی دانش و مهارت افراد است. در گذشته، آزمون‌ها به صورت دستی برگزار می‌شدند که این امر مشکلاتی از جمله اتلاف وقت، هزینه بالا و خطاهای انسانی را به همراه داشت. با پیشرفت فناوری، سیستم‌های آزمون‌گیری تحت سیستم به وجود آمدند که این مشکلات را تا حد زیادی برطرف کرده‌اند.

### ۱-۱ اجزای سیستم برگزاری آزمون

سیستم‌های آزمون‌گیری تحت سیستم از اجزای مختلفی تشکیل شده‌اند که از جمله آن‌ها می‌توان به موارد زیر اشاره کرد:

- **صفحه ورود کاربر:** این صفحه امکان ورود کاربر به سیستم را فراهم می‌کند.
- **\*صفحه مدیریت آزمون:** این صفحه امکان مدیریت آزمون‌ها از جمله ایجاد، ویرایش و حذف آزمون‌ها را فراهم می‌کند.
- **صفحه مدیریت سوالات:** این صفحه امکان مدیریت سوالات از جمله ایجاد، ویرایش و حذف سوالات را فراهم می‌کند.
- **صفحه برگزاری آزمون:** این صفحه امکان برگزاری آزمون برای کاربران را فراهم می‌کند.

### ۱-۱-۱ مدیریت آزمون

مدیریت آزمون در سیستم‌های آزمون‌گیری تحت سیستم شامل مراحل زیر است:

- **ایجاد آزمون:** در این مرحله، کاربر اطلاعات مربوط به آزمون را وارد می‌کند. از جمله این اطلاعات می‌توان به عنوان آزمون، مدت زمان آزمون، تعداد سوالات آزمون و نوع سوالات آزمون اشاره کرد.
- **ویرایش آزمون:** در این مرحله، کاربر می‌تواند اطلاعات آزمون را ویرایش کند.
- **حذف آزمون:** در این مرحله، کاربر می‌تواند آزمون را حذف کند.

## ۱-۱-۲ مدیریت سوالات

مدیریت سوالات در سیستم‌های آزمون‌گیری تحت سیستم شامل مراحل زیر است:

- ایجاد سوال: در این مرحله، کاربر سوالات آزمون را ایجاد می‌کند. از جمله اطلاعات مربوط به سوال می‌توان به عنوان سوال، متن و عکس سوال، پاسخ صحیح سوال و پاسخ‌های غلط سوال اشاره کرد.
- ویرایش سوال
- حذف سوال

## ۱-۱-۴ گزارش‌گیری

سیستم‌های آزمون‌گیری تحت سیستم امکان گزارش‌گیری دقیق از نتایج آزمون را فراهم می‌کنند. از جمله گزارش‌هایی که می‌توان از سیستم‌های آزمون‌گیری تحت سیستم دریافت کرد، می‌توان به موارد زیر اشاره کرد:

- گزارش نتایج آزمون: این گزارش شامل اطلاعات مربوط به عملکرد هر کاربر در آزمون است.
- گزارش کلی آزمون: این گزارش شامل اطلاعات کلی مربوط به آزمون از جمله تعداد شرکت‌کنندگان در آزمون، میانگین نمره آزمون و درصد قبولی در آزمون است.

## ۱-۱-۵ نتیجه‌گیری

سیستم‌های آزمون‌گیری تحت سیستم مزایای زیادی نسبت به آزمون‌گیری دستی دارند. این سیستم‌ها می‌توانند در بسیاری از زمینه‌های مختلف از جمله آموزش، استخدام و ارزیابی عملکرد مورد استفاده قرار گیرند. آزمون‌گیری یکی از روش‌های مهم ارزیابی دانش و مهارت افراد است. در گذشته، آزمون‌ها به صورت دستی برگزار می‌شدند که این امر مشکلاتی از جمله اتلاف وقت، هزینه بالا و خطاهای انسانی را به همراه داشت. با پیشرفت فناوری، سیستم‌های آزمون‌گیری تحت سیستم به وجود آمدند که این مشکلات را تا حد زیادی برطرف کرده‌اند.

## ۱-۲ مزایای استفاده از سیستم آزمون‌گیر

سیستم‌های آزمون‌گیری تحت سیستم با استفاده از زبان‌های برنامه‌نویسی، پایگاه‌های داده و نرم‌افزارهای طراحی واسط کاربر ساخته می‌شوند. این سیستم‌ها مزایای زیادی نسبت به آزمون‌گیری دستی دارند که از جمله آن‌ها می‌توان به موارد زیر اشاره کرد:



- صرفه‌جویی در زمان و هزینه: آزمون‌گیری تحت سیستم به دلیل خودکار بودن، زمان و هزینه کمتری نسبت به آزمون‌گیری دستی نیاز دارد.
- کاهش خطاهای انسانی: آزمون‌گیری تحت سیستم به دلیل حذف دخالت انسان، خطاهای انسانی را به حداقل می‌رساند.
- قابلیت انعطاف‌پذیری: آزمون‌های تحت سیستم را می‌توان با توجه به نیازهای مختلف، به راحتی شخصی‌سازی کرد.
- امکان گزارش‌گیری: آزمون‌های تحت سیستم امکان گزارش‌گیری دقیق از نتایج آزمون را فراهم می‌کنند.

در فصل بعدی، قرار است در مورد تحلیل سیستم و پیاده‌سازی نمودارهای مورد نیاز برای این پروژه صحبت شود.

در تحلیل سیستم، ابتدا باید سیستم مورد نظر را به اجزای کوچک‌تر تقسیم کرد. این اجزا می‌توانند اجزای فیزیکی، منطقی یا عملکردی باشند. به عنوان مثال، یک سیستم آزمون‌گیری تحت سیستم را می‌توان به اجزای زیر تقسیم کرد:

- اجزای فیزیکی: مانند رایانه، نرم‌افزار، پایگاه داده و دستگاه‌های ورودی/خروجی
- اجزای منطقی: مانند الگوریتم‌های پردازش سوالات، تصحیح آزمون و گزارش‌گیری
- اجزای عملکردی: مانند ایجاد آزمون، ویرایش آزمون، برگزاری آزمون و گزارش‌گیری

پس از تقسیم سیستم به اجزای کوچک‌تر، باید عملکرد هر یک از این اجزا مورد بررسی قرار گیرد. این بررسی می‌تواند شامل موارد زیر باشد:

- هدف هر جزء چیست؟
- چگونه کار می‌کند؟
- چه ورودی‌هایی دریافت می‌کند؟
- چه خروجی‌هایی تولید می‌کند؟
- چه محدودیت‌هایی دارد؟

تحلیل سیستم به شما کمک می‌کند تا درک بهتری از سیستم مورد نظر داشته باشید و نیازهای آن را شناسایی کنید. این اطلاعات می‌تواند برای طراحی و پیاده‌سازی سیستم مورد استفاده قرار گیرد.

در فصل بعدی، باید موارد زیر را در مورد تحلیل سیستم در نظر بگیرید:

- هدف از تحلیل سیستم چیست؟
- سیستم مورد نظر را چگونه به اجزای کوچکتر تقسیم خواهید کرد؟
- عملکرد هر یک از اجزای سیستم را چگونه بررسی خواهید کرد؟
- در پایان تحلیل سیستم، چه اطلاعات و نتایجی خواهید داشت؟

### پیاده‌سازی نمودارها

در این پروژه، از نمودارهای زیر برای نمایش اطلاعات سیستم استفاده خواهد شد:

- نمودار کلاس: این نمودار برای نمایش ساختار کلاس‌های برنامه استفاده می‌شود.
  - نمودار uml: این نمودار برای نمایش روابط بین کلاس‌های برنامه استفاده می‌شود.
  - نمودار جریان داده: این نمودار برای نمایش جریان داده در سیستم استفاده می‌شود.
- در فصل بعدی، باید موارد زیر را در مورد پیاده‌سازی نمودارها در نظر بگیرید:

- نمودارهای مورد نیاز برای این پروژه کدامند؟
- نحوه پیاده‌سازی هر یک از نمودارها چگونه است؟
- اطلاعات مورد نیاز برای پیاده‌سازی هر یک از نمودارها چیست؟

با پاسخگویی به این سوالات، می‌توان نمودارهای مورد نیاز برای این پروژه را به درستی پیاده‌سازی کرد.

در ادامه، به برخی از نکات مهم در مورد تحلیل سیستم و پیاده‌سازی نمودارها اشاره می‌کنیم:

- تحلیل سیستم باید به صورت دقیق و کامل انجام شود تا نیازهای سیستم به درستی شناسایی شوند.
- نمودارها باید به صورت واضح و مختصر طراحی شوند تا اطلاعات مورد نیاز به راحتی در آن‌ها قابل فهم باشند.
- در هنگام پیاده‌سازی نمودارها، باید از استانداردهای مربوطه پیروی شود.

با رعایت این نکات، می‌توانید تحلیل سیستم و پیاده‌سازی نمودارها را به درستی انجام دهید و اطلاعات مورد نیاز را به صورت دقیق و مختصر در اختیار کاربران قرار دهید.

## فصل دوم: تحلیل سیستم آزمون گیر

### ۲-۱ تجزیه و تحلیل سیستم آزمون گیر

برای تحلیل سیستم آزمون گیر و مدیریت آزمون، مراحل زیر دنبال شده است:

#### ۲-۱-۱ جمع آوری نیازمندی‌ها

جمع آوری نیازمندی‌ها اولین مرحله از فرآیند طراحی یک سیستم آزمون گیر است. در این مرحله، نیازهای کاربران سیستم شناسایی و مستند می‌شوند. این نیازها شامل موارد زیر است:

- نیازهای عملکردی: این نیازها مشخص می‌کنند که سیستم باید چه کاری انجام دهد. به عنوان مثال، سیستم باید بتواند آزمون‌های جدید ایجاد کند، سوالات آزمون را مدیریت کند، کارنامه داوطلبان را صادر کند و غیره.
  - نیازهای غیر عملکردی: این نیازها مشخص می‌کنند که سیستم باید با چه کیفیتی کار کند. به عنوان مثال، سیستم باید از امنیت بالایی برخوردار باشد، قابلیت استفاده بالایی داشته باشد و غیره.
- جمع آوری نیازمندی‌ها معمولاً با مصاحبه با کاربران سیستم، برگزاری جلسات پرسش و پاسخ و مطالعه اسناد و مدارک موجود انجام می‌شود. در این مرحله، مهم است که تمام نیازهای کاربران شناسایی شوند. هرچه نیازهای بیشتری شناسایی شود، سیستم طراحی شده دقیق‌تر و کارآمدتر خواهد بود.
- در ادامه، مراحل جمع آوری نیازمندی‌ها برای طراحی یک سیستم آزمون گیر به صورت خلاصه توضیح داده می‌شود:

- شناسایی کاربران: اولین قدم، شناسایی کاربران سیستم است. کاربران سیستم می‌توانند شامل داوطلبان، کارمندان و مدیران باشند.
- تعیین اهداف سیستم: در این مرحله، اهداف سیستم تعیین می‌شوند. اهداف سیستم باید واضح و قابل اندازه‌گیری باشند.
- شناسایی نیازها: در این مرحله، نیازهای کاربران سیستم شناسایی می‌شوند. نیازها می‌توانند به صورت عملکردی یا غیر عملکردی باشند.
- مستند سازی نیازها: در این مرحله، نیازهای شناسایی شده مستند می‌شوند. مستند سازی نیازها به منظور تسهیل فرآیند طراحی و توسعه سیستم ضروری است.
- جمع آوری نیازمندی‌ها یک فرآیند مهم و ضروری است که باید با دقت انجام شود. هرچه نیازهای بیشتری شناسایی شود، سیستم طراحی شده دقیق‌تر و کارآمدتر خواهد بود.

## ۲-۱-۲ تحلیل نیازمندی‌ها

### ۲-۱-۲-۱ نیازمندی‌های کاربری:

#### داوطلبان:

- ثبت نام: امکان ثبت نام و ورود به سیستم با استفاده از کد ورود و رمز داوطلبی.
- مشاهده آزمون‌ها: دسترسی به لیست آزمون‌های موجود و جزئیات آن‌ها.
- شرکت در آزمون: امکان شرکت در آزمون و پاسخ به سوالات با رعایت زمان بندی و قوانین مربوطه.

#### کارمندان (مدیران):

- مدیریت آزمون‌ها: ایجاد، ویرایش، و حذف آزمون‌ها به همراه تنظیم تاریخ، تعداد سوالات و زمان آزمون.
- مدیریت سوالات: افزودن، ویرایش، و حذف سوالات از آزمون‌ها به همراه بارگذاری تصاویر و متن پاسخ صحیح.
- تعیین ارزش سوالات: قابلیت تعیین ارزش سوالات به صورت دستی یا اتوماتیک برای تشخیص نمره دهی.
- گزارش گیری: ایجاد گزارشات مرتبط با آزمون‌ها، سوالات، و داوطلبان.

### ۲-۲-۱-۲ نیازمندی‌های فنی:

- امنیت: اطمینان از امنیت اطلاعات داوطلبان و جلوگیری از دسترسی غیرمجاز به سیستم.
- کارایی: اجرای سریع و بهینه آزمون‌ها، مدیریت دقیق زمان و تعداد سوالات.
- قابلیت اطمینان: جلوگیری از دست رفتن داده‌ها و ایجاد پشتیبان‌های قابل بازیابی.

### ۲-۲-۱-۳ نیازمندی‌های عملیاتی:

- اطلاع رسانی: ارسال اطلاعیه‌ها و اطلاعات مربوط به آزمون‌ها به داوطلبان و کارمندان.

- \*\*\*مدیریت کاربران:\*\*\* امکان ایجاد، ویرایش، و حذف کاربران جدید و تعیین سطح دسترسی آن‌ها.

این تحلیل اولیه به معرفی نیازمندی‌های اساسی سیستم آزمون گیر می‌پردازد. این نیازمندی‌ها می‌توانند در مراحل بعدی با جزئیات بیشتر و با توجه به نیازهای ویژه کاربران و محیط سیستم گسترش یابند.

## ۲-۱-۳ طراحی سیستم:

پس از مرحله تجزیه و تحلیل نیازمندی‌ها که نیازمندی‌های کاربری، فنی و عملیاتی سیستم شناسایی و مستند شدند نوبت به طراحی سیستم می‌رسد تا با استفاده از مستندات به طراحی سیستم پرداخت.

مراحل طراحی سیستم آزمون گیر را بر اساس نیازمندی‌های کاربری، فنی و عملیاتی ارائه می‌دهد. این مراحل شامل موارد زیر است:

- طراحی ساختار کلی سیستم
- طراحی سیستم
- مستندسازی
- تست و ارزیابی

## ۲-۱-۳-۱ طراحی ساختار کلی سیستم

در این مرحله، اجزای اصلی سیستم و روابط بین آن‌ها شناسایی و مستند می‌شوند. این اجزا می‌تواند شامل موارد زیر باشد:

- داوطلبان: کاربرانی که در آزمون شرکت می‌کنند.
- آزمون‌ها: مجموعه سوالاتی که برای ارزیابی دانش و مهارت داوطلبان طراحی شده‌اند.
- سوالات: پرسش‌هایی که در آزمون مطرح می‌شوند.
- مدیران: کاربرانی که مسئول مدیریت سیستم هستند.

همچنین، در این مرحله، عملکردهای اصلی کاربران در سیستم شناسایی و مستند می‌شوند. این عملکردها می‌تواند شامل موارد زیر باشد:

- ثبت‌نام: داوطلبان برای شرکت در آزمون باید در سیستم ثبت‌نام کنند.
- مدیریت آزمون‌ها: مدیران می‌توانند آزمون‌های جدید ایجاد کنند، سوالات آزمون‌ها را مدیریت کنند و گزارش‌های مربوط به آزمون‌ها را دریافت کنند.
- شرکت در آزمون: داوطلبان می‌توانند در آزمون‌های موجود شرکت کنند.
- گزارش‌گیری: مدیران می‌توانند گزارش‌های مختلفی را از سیستم دریافت کنند.

### طراحی سیستم

در این مرحله، طراحی سیستم به صورت جزئی‌تر انجام می‌شود. در این مرحله، نمودارهای UML، طراحی پایگاه داده و طراحی رابط کاربری انجام می‌شوند.

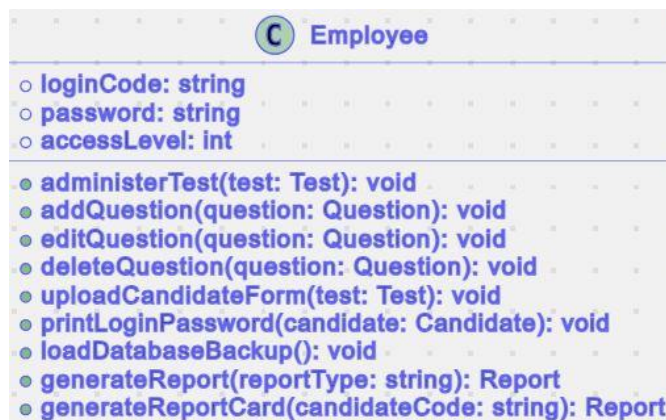
### الف) نمودارهای UML

نمودارهای UML ابزاری قدرتمند برای مستندسازی و طراحی سیستم‌های نرم‌افزاری هستند. در طراحی سیستم آزمون‌گیر، از نمودارهای زیر استفاده می‌شود:

**الف-۱) Class Diagram:** این نمودار، اجزای مختلف سیستم و روابط بین آن‌ها را نشان می‌دهد.

UML به عنوان یک ابزار استاندارد در مهندسی نرم‌افزار به منظور نمایش ساختار کلاس‌ها و ارتباطات آن‌ها استفاده می‌شود. نماد "C" درون یک دایره به عنوان نماینده‌ای از یک کلاس در UML شناخته می‌شود.

قبل از طراحی نمودار class diagram باید با ساختارهای هر کلاس آشنا شده و سپس نمودار کلی را رسم نمود:



شکل (۲-۱) نمودار کلاس کارمند

تصویر نمودار کلاس UML برای کلاس "کارمند" شامل ویژگی‌ها و عملیات‌های مربوطه است. ویژگی‌های این کلاس شامل سه ویژگی اصلی است: "loginCode" که از نوع رشته (String) است، "گذرواژه" نیز از نوع رشته (String) و "accessLevel" که از نوع عدد صحیح (int) می‌باشد.

عملیات‌های مربوط به این کلاس شامل عملیات‌های مختلفی است که می‌تواند توسط کارمندان انجام شود. این عملیات شامل موارد زیر می‌باشد:

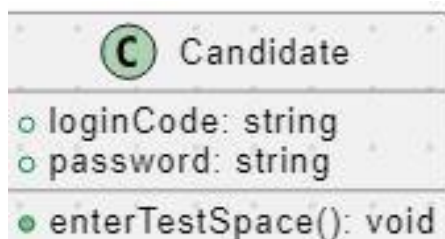
- \*\*برگزاری آزمون:\*\* یک عملیات که برای برگزاری آزمون در سیستم استفاده می‌شود.
- \*\*افزودن، ویرایش و حذف سؤالات:\*\* عملیاتی که به کارمندان اجازه می‌دهد سؤالات را به آزمون اضافه کنند، آن‌ها را ویرایش کنند یا از آزمون حذف کنند.
- \*\*آپلود فرم‌های داوطلب:\*\* یک عملیات برای بارگذاری فرم‌های داوطلب برای آزمون‌ها.
- \*\*چاپ رمز ورود به سیستم:\*\* عملیاتی که رمز ورود به سیستم را برای کاربران تولید می‌کند.
- \*\*بارگیری نسخه‌های پشتیبان پایگاه داده:\*\* یک عملیات که از کارمندان اجازه می‌دهد نسخه‌های پشتیبان از پایگاه داده را دانلود کنند.
- \*\*تولید گزارش و کارنامه:\*\* عملیاتی که گزارشات و کارنامه‌ها را برای کاربران تولید می‌کند.

علاوه بر این، متدهای این عملیات ممکن است انواع بازگشتی متفاوتی داشته باشند. مثلا:

- **void**: برای متدهایی که هیچ مقدار بازگشتی ندارند مانند برگزاری آزمون یا بارگیری نسخه پشتیبان.
- **Report**: برای متدهایی که نمونه‌ای از کلاس یا نوع "گزارش" را برمی‌گردانند، مانند تولید گزارش و کارنامه.

در کل، کلاس "کارمند" برای مدیریت و اجرای عملیات مختلفی که در یک سیستم آزمون ویرایش، گزارش‌گیری و سایر فعالیت‌های مرتبط با آزمون‌ها صورت می‌گیرد طراحی شده است.

۲-۱-۲ نمودار کلاس برای کارمند:



شکل (۲-۲) نمودار کلاس کارمند

این نمودار شامل ویژگی‌ها و عملیات‌های مربوط به این کلاس است:

• ویژگی‌ها:

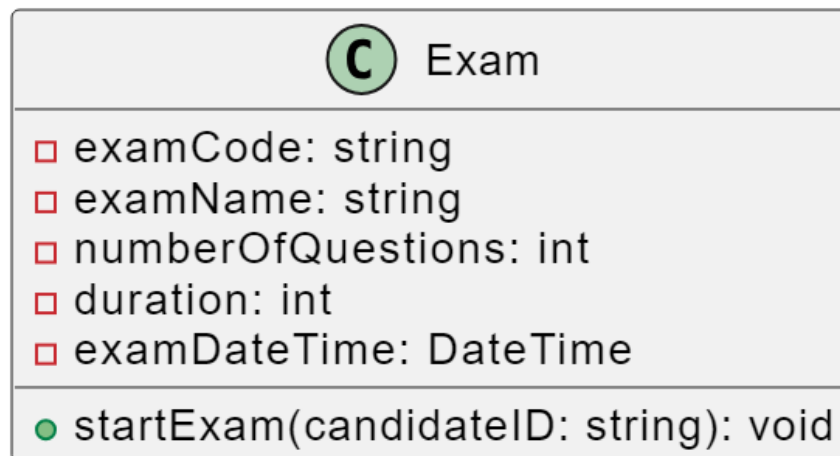
- **loginCode: string**: نشان دهنده ویژگی loginCode که از نوع رشته (string) است و اطلاعات کد ورود به سیستم را نمایش می‌دهد.
- **password: string**: ویژگی password نیز از نوع رشته (string) است و برای نگهداری رمز عبور داوطلب استفاده می‌شود.



• عملیات:

- `void**enterTestSpace(): void` نمایش تابع `enterTestSpace` که بدون بازگشت مقدار (`void`) است. این تابع مربوط به ورود داوطلب به فضای آزمون یا انجام عملیاتی مرتبط با آزمون است.

۲-۱-۳ نمودار کلاس برای آزمون:



شکل (۳-۲) نمودار کلاس آزمون

توضیحات ارائه شده در نمودار کلاس "Exam" به خوبی ساختار و عملکرد این کلاس را مشخص می‌کند. این نمودار یک مدل ساختاری از کلاس "Exam" است که در یک سیستم نرم‌افزاری برای مدیریت و اجرای آزمون‌ها استفاده می‌شود. بیایید این توضیحات را گسترش دهیم:

\*\*\*کلاس "Exam":\*\*\*

- کلاسی است که نماینده‌ای از هر آزمون در سیستم می‌باشد. این کلاس مسئولیت‌های مربوط به مدیریت و ایجاد آزمون را دارد.

**\*\*ویژگی‌های کلاس "Exam":\*\***

- **\*\*examCode\*\***: یک کد یکتا برای شناسایی آزمون.

- **\*\*examName\*\***: عنوان یا نام آزمون.

- **\*\*numberOfQuestions\*\***: تعداد سوالات موجود در آزمون.

- **\*\*duration\*\***: مدت زمان مجاز برای انجام آزمون.

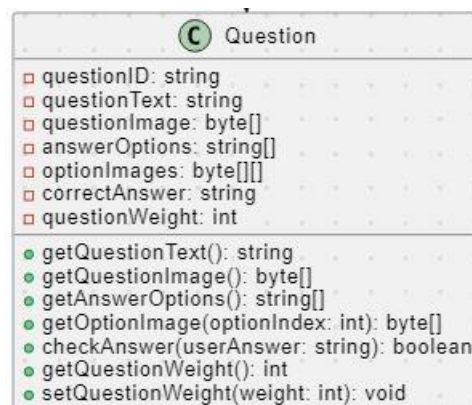
- **\*\*examDateTime\*\***: تاریخ و زمان برگزاری آزمون.

**\*\*عملیات کلاس "Exam":\*\***

- **\*\*startExam(candidateID: string): void\*\***: این عملیات با دریافت شناسه داوطلب، آغاز آزمون برای داوطلب مشخص شده را فراهم می‌کند.

این کلاس اساسی است و مسئولیت‌های اصلی مدیریتی و اجرایی آزمون را در سیستم بر عهده دارد. اطلاعات موجود در این نمودار برای طراحی و پیاده‌سازی قسمت‌های مربوط به آزمون‌ها در سیستم به کار می‌روند و توسعه‌دهندگان را در درک بهتر از ساختار آزمون‌ها کمک می‌کنند.

۲-۱-۳-۴ نمودار کلاس سوال:



شکل (۲-۴) نمودار کلاس سوال

این ویژگی‌ها و روش‌ها عملکردی را پیشنهاد می‌کنند که شامل ذخیره متن سؤال، یک تصویر مرتبط، گزینه‌های پاسخ (در متن و احتمالاً تصاویر) و پاسخ صحیح است. علاوه بر این، وزنی در رابطه با سؤال وجود دارد که می‌تواند بر امتیاز دهی تأثیر بگذارد. این روش‌ها قابلیت بازیابی این اطلاعات، بررسی صحیح بودن پاسخ ارائه شده توسط کاربر و تنظیم یا بازیابی وزن سؤال را فراهم می‌کنند.

با توجه به توضیحات ارائه شده در نمودار کلاس "سؤال"، این نمودار به خوبی ویژگی‌ها و روش‌های کلاس "سؤال" را نشان می‌دهد که برای مدل‌سازی ساختار و عملکرد سوالات مورد استفاده در مسابقات، آزمون‌ها یا نظرسنجی‌ها طراحی شده است.

**\*\*کلاس "سؤال"\*\*:**

- کلاسی است که معمولاً برای نگهداری و مدیریت سوالات مورد استفاده قرار می‌گیرد.

**\*\*ویژگی‌های کلاس "سؤال"\*\*:**

- **questionID**: رشته‌ای که برای شناسایی یکتای سؤال استفاده می‌شود.

- **questionText**: متن یا مطلب سؤال.

- **questionImage**: بایت‌های تصویر مرتبط با سؤال.

- **answerOptions**: لیست گزینه‌های پاسخ به صورت متنی.

- **optionImages**: آرایه‌ای از بایت‌های مرتبط با تصاویر گزینه‌های پاسخ.

- **correctAnswer**: پاسخ صحیح به سؤال.

- **questionWeight**: وزن یا اهمیت سؤال.

**\*\*عملیات کلاس "سؤال"\*\*:**

- **getText()**: برگرداندن متن سؤال.

- **getImage()**: برگرداندن تصویر سؤال به صورت بایت.

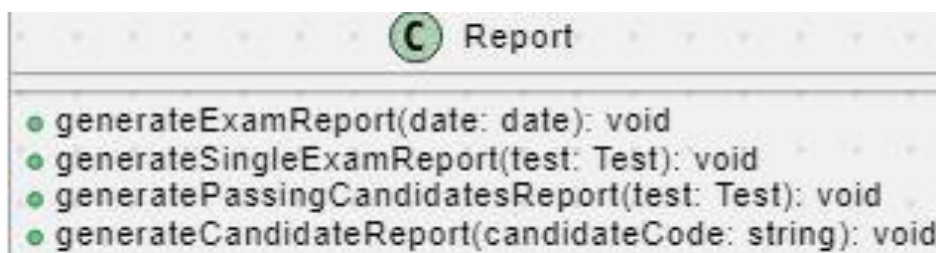
- **getAnswerOptions()**: برگرداندن لیست گزینه‌های پاسخ به صورت متنی.

- **getOptionImage(optionIndex: int)**: برگرداندن تصویر گزینه‌های پاسخ به صورت بایت.

- **checkAnswer(userAnswer: string)**: بررسی صحت پاسخ داده شده توسط کاربر.

- `getQuestionWeight()`: برگرداندن وزن سوال.
- `setQuestionWeight(weight: int)`: تنظیم وزن جدید برای سوال.

۲-۱-۳ نمودار کلاس برای گزارش:



شکل (۲-۵) نمودار کلاس گزارش

این قطعه کد یک توصیف از متدهای مربوط به گزارش‌دهی در یک سیستم نرم‌افزاری است. این متدها مربوط به ایجاد گزارش‌های مختلفی برای آزمون‌ها در سیستم هستند. هر یک از این متدها گزارش مشخصی را براساس ورودی‌های مختلفی ایجاد می‌کنند.

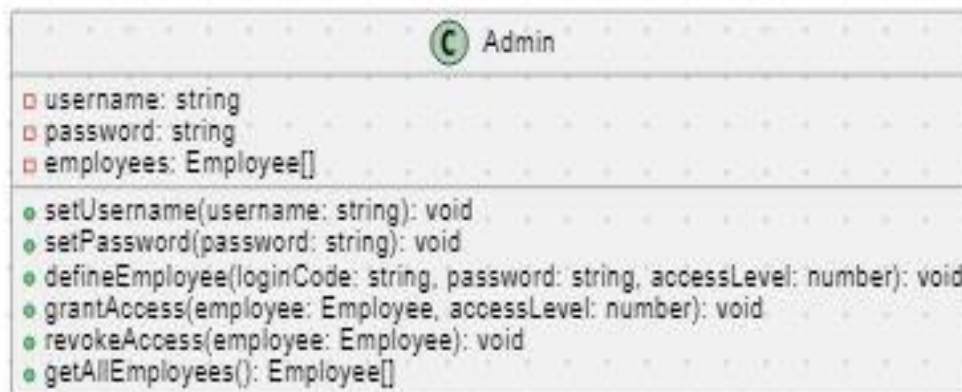
`generateExamReport(date: date): void`: این متد با گرفتن تاریخ به عنوان ورودی، احتمالاً یک گزارش کلی یا مجموعه‌ای از آزمون‌هایی که در آن تاریخ انجام شده‌اند، تولید می‌کند.

`generateSingleExamReport(test: Test): void`: این متد با گرفتن یک شیء تست به عنوان ورودی، احتمالاً یک گزارش جزئی و مشخصی از یک آزمون خاص را ایجاد می‌کند.

`generatePassingCandidatesReport(test: Test): void`: این متد با گرفتن یک شیء تست به عنوان ورودی، ممکن است یک گزارش در مورد داوطلبانی که در آزمون موردنظر قبول شده‌اند، ایجاد کند.

`generateCandidateReport(candidateCode: string): void` : این متد با گرفتن یک کد کاندید به عنوان ورودی، احتمالاً یک گزارش مربوط به یک داوطلب خاص را تولید می‌کند.

۲-۱-۳-۶ نمودار کلاس برای مدیر :



شکل (۲-۶) نمودار کلاس مدیر

. در این نمودار کلاس "Admin"، اطلاعات و عملیات مربوط به یک مدیر سیستم نرم‌افزاری نمایش داده شده است.

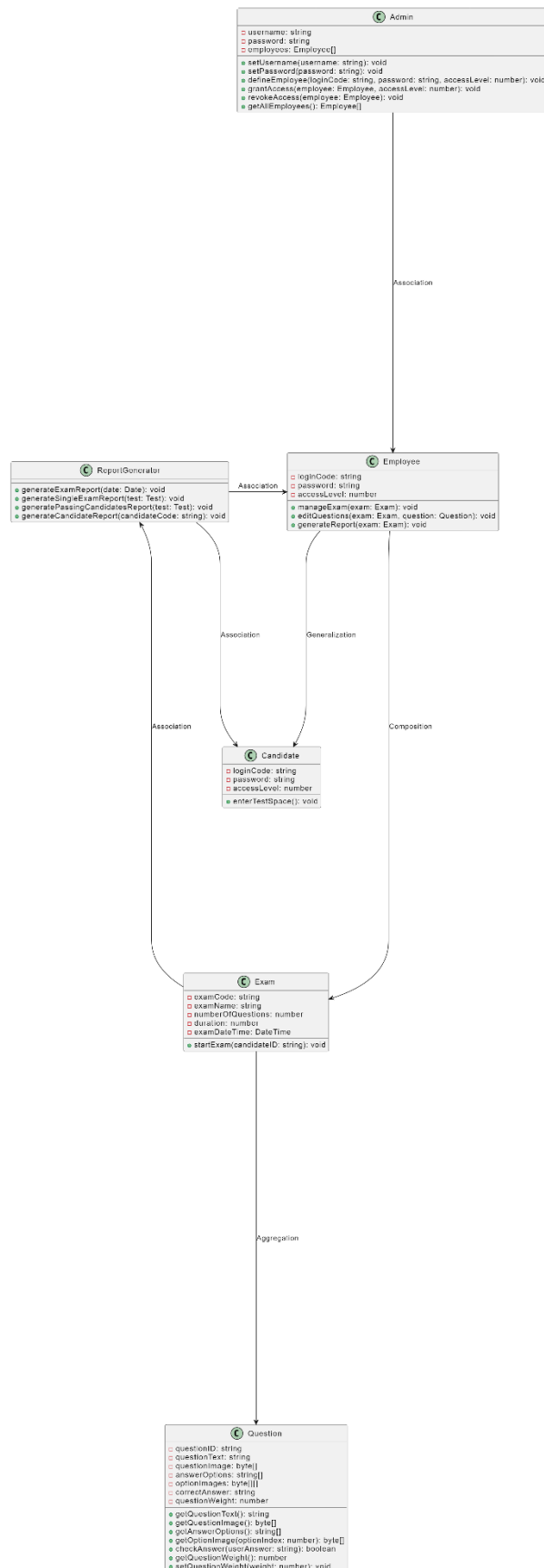
کلاس "Admin" در سیستم نرم‌افزاری ما مسئولیت‌های مهمی دارد. به عنوان مثال، می‌تواند اطلاعات ورود به سیستم را مدیریت کرده و دسترسی‌ها را به کارمندان اختصاص دهد یا لغو کند. ویژگی‌های متنوعی مانند نام کاربری و رمز عبور برای احراز هویت افراد استفاده می‌شوند. این اطلاعات مهم برای ارتباط و ورود به سیستم از اهمیت ویژه‌ای برخوردارند.

در این کلاس دیگرام، توابع یا متدهایی نیز تعریف شده‌اند که وظایف مختلفی انجام می‌دهند، مانند تنظیم نام کاربری یا رمز عبور، تعریف کارمندان جدید، تخصیص و لغو دسترسی‌ها و همچنین دریافت لیست کارمندان. این متدها برای مدیریت اطلاعات و دسترسی‌های سیستم به کار می‌روند.

استفاده از این کلاس در سیستم ما به ما امکان می‌دهد که یک نقش مدیریتی و قابل اعتماد را برای کنترل و مدیریت کلیه فعالیت‌ها در سیستم به صورت منظم و سازمان‌یافته داشته باشیم. این اطلاعات می‌توانند به‌طور کامل در مستندات پروژه نوشته شوند تا توضیح دقیقی از وظایف و نقش این کلاس در ساختار کلی سیستم فراهم شود.



## ۳-۱-۲ نمودار کلاس دیاگرام سیستم مدیریت آزمون



شکل (۲-۷) نمودار کلاس سیستم آزمون گیری

پس از آشنایی با اجزای کلاس های مختلف سیستم آزمون گیر با استفاده از روابط موجود میان کلاس های uml این ارتباطات را پیاده سازی کردیم :

با توجه به روابط بین کلاس ها در دیاگرام سیستم آزمون گیر، می توان به نتیجه گیری های زیر رسید:

#### **\*\*Admin -----> Employee : Association\*\***

- این رابطه نشان می دهد که کلاس "Admin" ممکن است با کلاس "Employee" ارتباط داشته باشد. این ارتباط ممکن است برای استفاده از اطلاعات یا سرویس های ارائه شده توسط کارمندان بکار رود.

#### **\*\*Employee -----> Candidate : Generalization\*\***

- این نشان می دهد که "Employee" به عنوان یک نوع ویژه از "Candidate" مدل سازی شده است. این ممکن است به معنای این باشد که هر کارمند همزمان می تواند به عنوان یک داوطلب نیز عمل کند.

#### **\*\*Employee -----> Exam : Composition\*\***

- این رابطه Composition نشان می دهد که یک "Employee" می تواند "Exam" ها را ایجاد و مدیریت کند. معمولاً Composition به معنای ارتباط سریع تر و نزدیک تر از ساده ترین Association است.

#### **\*\*Exam -----> Question : Aggregation\*\***

- این Aggregation نشان می دهد که "Exam" شامل "Question" ها می شود. به این معناست که یک آزمون می تواند شامل چندین سوال باشد ولی سوالات به طور مستقل از آزمون نیز می توانند وجود داشته باشند.



**\*\*Exam -----> ReportGenerator : Association\*\***

- این رابطه Association نشان می‌دهد که یک "Exam" ممکن است با "ReportGenerator" برای تولید گزارشات ارتباط داشته باشد، مثلاً گزارش‌های نتایج آزمون‌ها.

**\*\*Question -----> (No specific relationship with other classes)\*\***

- کلاس "Question" هیچ ارتباط خاصی با سایر کلاس‌ها ندارد، احتمالاً به دلیل استقلال و مستقلیت بیشتر سوالات در محیط آزمونی.

**\*\*Candidate -----> (No specific relationship with other classes)\*\***

- کلاس "Candidate" نیز ارتباط خاصی با سایر کلاس‌ها ندارد، احتمالاً به دلیل استقلال و مستقلیت داوطلبان در فرآیند آزمون.

**\*\*ReportGenerator -> Exam : Association\*\***

- این Association نشان می‌دهد که "ReportGenerator" ممکن است با "Exam" برای تولید گزارشات ارتباط داشته باشد.

**\*\*ReportGenerator -> Candidate : Association\*\***

- این Association نشان می‌دهد که "ReportGenerator" ممکن است با "Candidate" برای تولید گزارشات ارتباط داشته باشد.

**\*\*ReportGenerator -> Employee : Association\*\***



- این Association نشان می‌دهد که "ReportGenerator" ممکن است با "Employee" برای تولید گزارشات ارتباط داشته باشد.

این روابط نشان‌دهنده تعاملات و ارتباطات میان کلاس‌های مختلف در سیستم آزمون گیر هستند. استفاده از این روابط و ارتباطات به منظور ارتباط دقیق بین کلاس‌ها و توصیف کامل عملکرد سیستم بسیار مفید است.

## الف - ۲) Sequence Diagram

نمودار دنباله‌ای یا Sequence Diagram در زبان مدلسازی UML به منظور نشان دادن ترتیب زمانی عملکرد متعددی که بین اشیاء (Objects) در یک سیستم رخ می‌دهد، استفاده می‌شود. این نمودار به خوبی دنبال می‌کند که چگونه اشیاء یکدیگر را فراخوانی می‌کنند و پاسخ می‌دهند.

در اصل، Sequence Diagram یک نمایش چشم‌گیر از رفتار و تبادل پیام بین اشیاء را فراهم می‌کند. این نمودار با استفاده از لاین‌های عمودی برای نشان دادن اشیاء و چیدمان پیغام‌ها بین این اشیاء از چپ به راست (زمانی)، به‌طور دقیق نشان می‌دهد که هر شیء چه زمانی با چه شیء دیگری ارتباط داشته و پیام چه نوع اطلاعات یا درخواستی را به آن می‌فرستد.

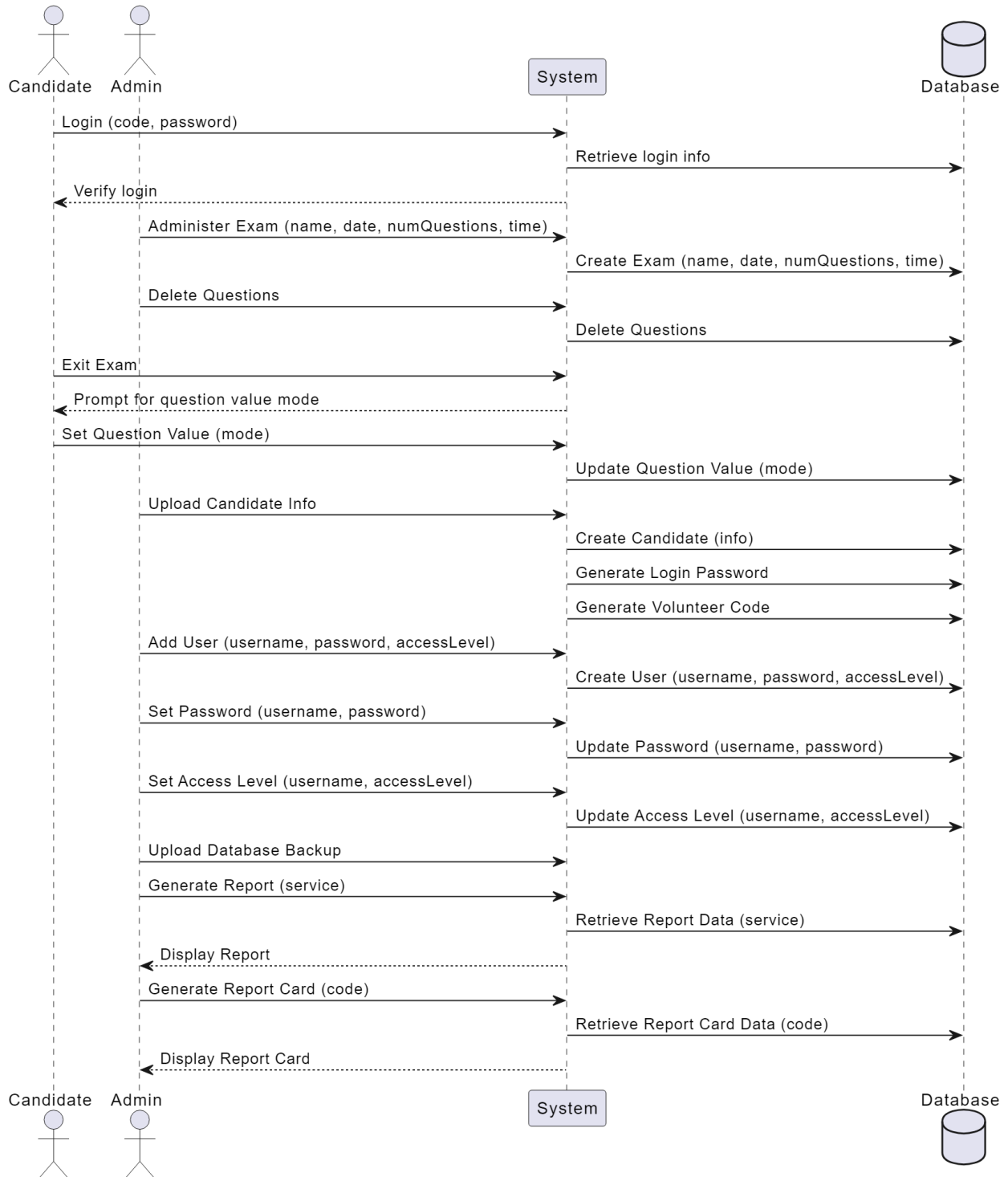
توضیحات دقیق از چگونگی انجام یک فرایند و مبادلات پیام بین اشیاء، جریان کار را به شکل گرافیکی به نمایش می‌گذارد و به توسعه‌دهندگان و تیم‌های مختلف سیستم‌ها کمک می‌کند تا فرآیندهای سیستم خود را بهتر درک کنند و از این طریق بهینه‌سازی و بهبود آن‌ها بپردازند.

۲-۱-۳-۷ نمودار دنباله‌ای سیستم مدیریت آزمون

واحد نجف آباد

شکل (۷-۲) نمودار دنباله ای  
(Sequence Diagram)

سیستم آزمون گیر



این دیاگرام جریان یک سیستم نرم‌افزاری را نشان می‌دهد که ارتباطی بین چندین شیء یا انتیتی (Candidate, Admin, System و Database) در یک سیستم مدیریت آزمون یا پلتفرم آزمون‌دهی را نمایش می‌دهد. این دیاگرام چند مرحله اصلی اجرای یک فرآیند را نشان می‌دهد:

۱. \*\*ورود Candidate و Admin\*\* با مرحله "ورود (کد، رمز عبور)" که به سیستم متصل می‌شود.
۲. سیستم پردازش‌هایی انجام می‌دهد مانند "تأیید ورود" و "بازیابی اطلاعات ورودی" از پایگاه داده و سپس به مرحله تأیید باز می‌گردد.
۳. Admin دارای چندین عملکرد است که شامل ایجاد آزمون، حذف سوالات، خروج از آزمون، نمایش حالت ارزیابی سوال، تنظیم ارزش سوال، بارگذاری اطلاعات کاندید، اضافه کردن کاربر، تنظیم رمز عبور، تنظیم سطح دسترسی، بارگذاری پشتیبان پایگاه داده، تولید گزارش و تولید کارت نمره می‌شود.
۴. هر یک از عملکردهای Admin متناظر با عملیاتی است که سیستم می‌تواند در پایگاه داده انجام دهد، مانند ایجاد آزمون، حذف سوالات، به‌روزرسانی ارزش سوال و ...
۵. پس از تولید گزارش یا کارت نمره، سیستم می‌تواند گزارش را نمایش دهد و یا بازخوردی را برای تولید گزارش یا کارت نمره دیگر ارائه دهد.

آیکون‌های مربوط به دو شخص (Candidate و Admin) و پایگاه داده در سمت چپ، راست و پایین این دیاگرام قرار دارند. پیکان‌ها جهت جریان داده یا دنباله عملیات بین این انتیتی‌ها را نشان می‌دهند. استفاده از خطوط مستقیم با پیکان‌ها جریان اصلی عملکرد را نشان می‌دهد و خطوط خط چین نشان دهنده بازگشت اطلاعات یا بازخورد است. استفاده از بخش‌های مختلف و ترتیب آن‌ها به درک روش عملکرد انتیتی‌ها و تعاملات آن‌ها با سیستم و پایگاه داده کمک می‌کند.

### الف - ۳) use case diagram

نمودار Use Case یا نمودار مورد استفاده یکی از مفیدترین ابزارهای مدل سازی است که در مهندسی نرم افزار استفاده می شود. این نمودار به طراحان نرم افزار کمک می کند تا نیازها و عملکردهایی که کاربران یا سیستم از سیستم می خواهند را شناسایی و مدل سازی کنند.

نمودار Use Case از دو مولفه اصلی تشکیل شده است:

**Use Cases (موارد استفاده):** این موارد نمایش می دهند که چه عملیاتی توسط کاربران یا سیستم انجام می شود. هر مورد استفاده یک واحد کارکردی مهم در سیستم را نشان می دهد که به نیازهای یک یا چند نقش در سیستم پاسخ می دهد. مثلاً، "ثبت نام"، "ورود به سیستم"، "ایجاد آزمون"، و "تولید گزارش" می توانند موارد استفاده باشند.

**Actors (نقش ها):**

این موارد نشان دهنده کاربران یا سیستم های خارجی هستند که با سیستم ارتباط دارند. یک نقش ممکن است یک کاربر واقعی مثل "داوطلب" یا یک سیستم خارجی مثل "پایگاه داده" یا "سیستم پردازش پرداخت" باشد. این نمودار Use Case برای سیستم آزمون گیر چگونگی تعامل بین کاربران و سیستم را نشان می دهد. موارد زیر می توانند جزئیات توصیفی این نمودار باشند:

#### ۲-۱-۳-۸ نمودار دنباله ای سیستم مدیریت آزمون

**کاربران:** سه کاربر اصلی در این سیستم حضور دارند:

- **کارمند (Employee):** کارمندان می توانند سطح دسترسی و مدیریت سیستم را تعیین کرده و یا وارد سیستم شوند.

- **مدیر (Admin):** مسئولیت های مدیریتی مختلفی از جمله ایجاد آزمون، حذف سوالات، تولید گزارشات و... دارند.

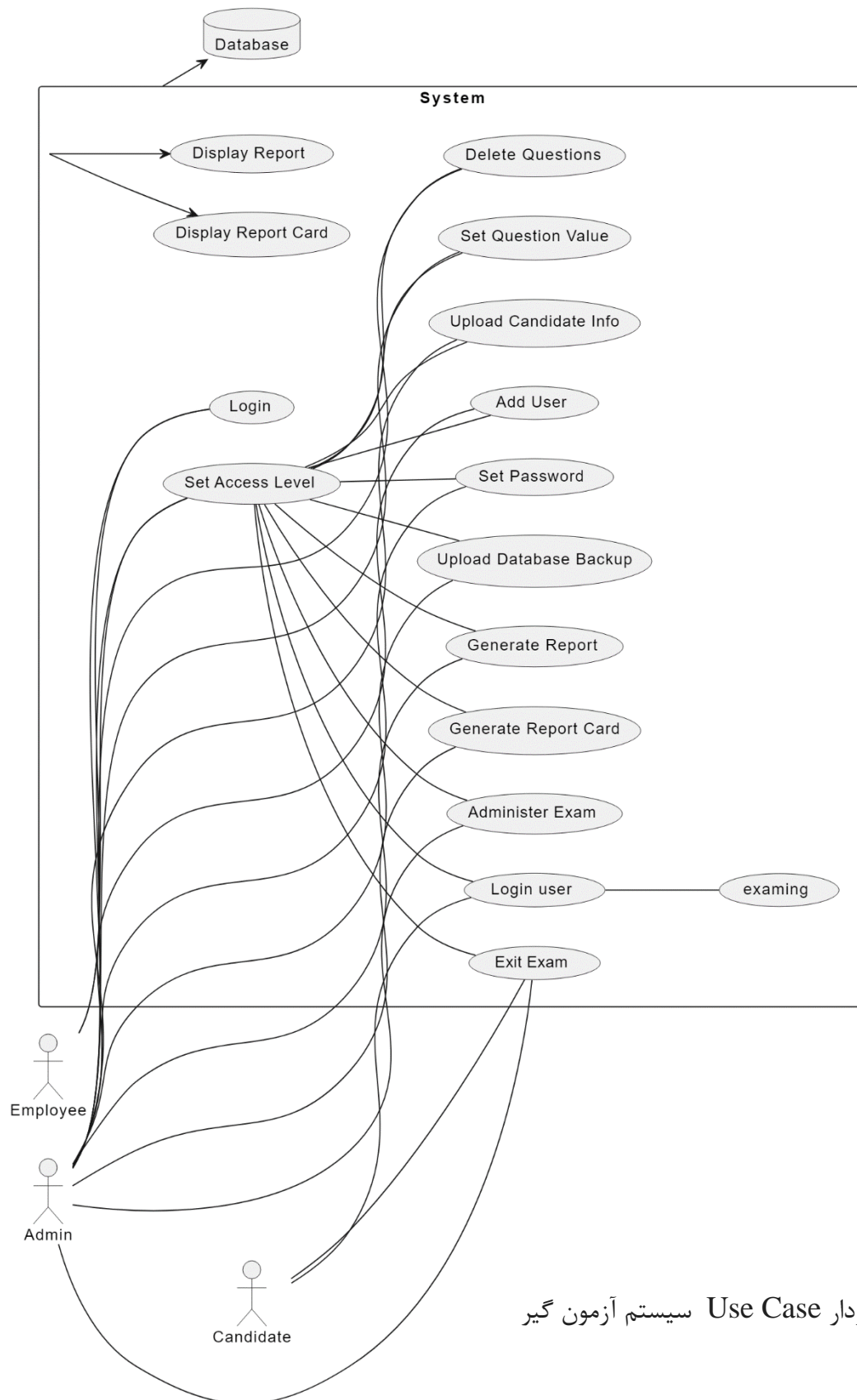
- **داوطلب (Candidate):** داوطلب ها می توانند وارد سیستم شده و آزمون را آغاز و به آن پاسخ دهند.

## \*\*\* Use Case ها \*\*\*

- \*\*\*Login\*\*\*: تمام کاربران باید بتوانند وارد سیستم شوند.
- \*\*\*Administer Exam\*\*\*: فقط مدیر مجاز به ایجاد و مدیریت آزمون‌ها است.
- \*\*\*Delete Questions\*\*\*: حذف سوالات تنها توسط مدیر انجام می‌شود.
- \*\*\*Exit Exam\*\*\*: این امکان به کاندیدها داده می‌شود تا از آزمون خارج شوند.
- \*\*\*Set Question Value\*\*\*: تنظیم وزن (اهمیت) سوالات توسط مدیر امکان‌پذیر است.
- \*\*\*Upload Candidate Info\*\*\*: قابلیت آپلود اطلاعات کاندیدها توسط مدیر وجود دارد.
- \*\*\*Add User\*\*\*: افزودن کاربر جدید توسط مدیر امکان‌پذیر است.
- \*\*\*Set Password\*\*\*: تنظیم رمز عبور توسط مدیر قابل انجام است.
- \*\*\*Set Access Level\*\*\*: تنظیم سطح دسترسی کارمندان توسط مدیر امکان‌پذیر است.
- \*\*\*Upload Database Backup\*\*\*: مدیران می‌توانند پشتیبان پایگاه داده را آپلود کنند.
- \*\*\*Generate Report Card و Generate Report\*\*\*: تولید گزارشات توسط مدیر امکان‌پذیر است.
- \*\*\*Display Report Card و Display Report\*\*\*: نمایش گزارشات توسط سیستم.

## \*\*\* رابطه بین Use Case ها \*\*\*

- مدیر (Admin) می‌تواند دسترسی (Set Access Level) را برای کارمندان تنظیم کند.
- مدیر (Admin) می‌تواند کارمند جدید تعریف کند.
- مدیر دسترسی به اکثر عملیات‌های مدیریتی دارد و با دسترسی اعطا شده می‌تواند به سایر عملیات نیز دسترسی داشته باشد.
- همچنین داوطلب می‌تواند وارد سیستم شده و سوالات آزمون را حل کند و همچنین از آزمون خارج شود



شکل (۸-۲) نمودار Use Case سیستم آزمون گیر

## ب ( طراحی پایگاه داده

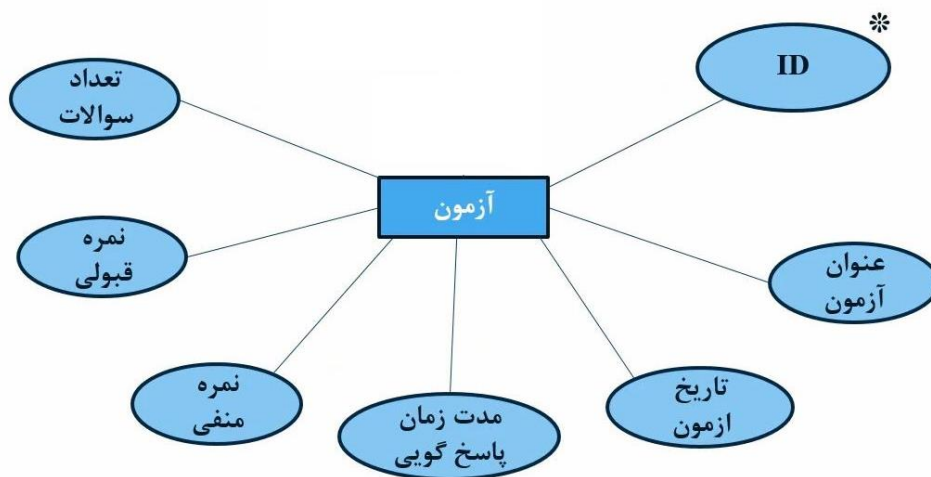
پایگاه داده، محل ذخیره‌سازی داده‌های مختلف سیستم است. در طراحی پایگاه داده، موجودیت‌ها، روابط و جداول مورد نیاز برای ذخیره‌سازی داده‌های مختلف سیستم شناسایی و طراحی می‌شوند.

اولین گام برای طراحی یک پایگاه داده شناخت موجودیت‌های سیستمی است که می‌خواهیم طراحی کنیم :

ب-۱ موجودیت ها :

۹-۳-۱-۲ موجودیت آزمون

### موجودیت آزمون



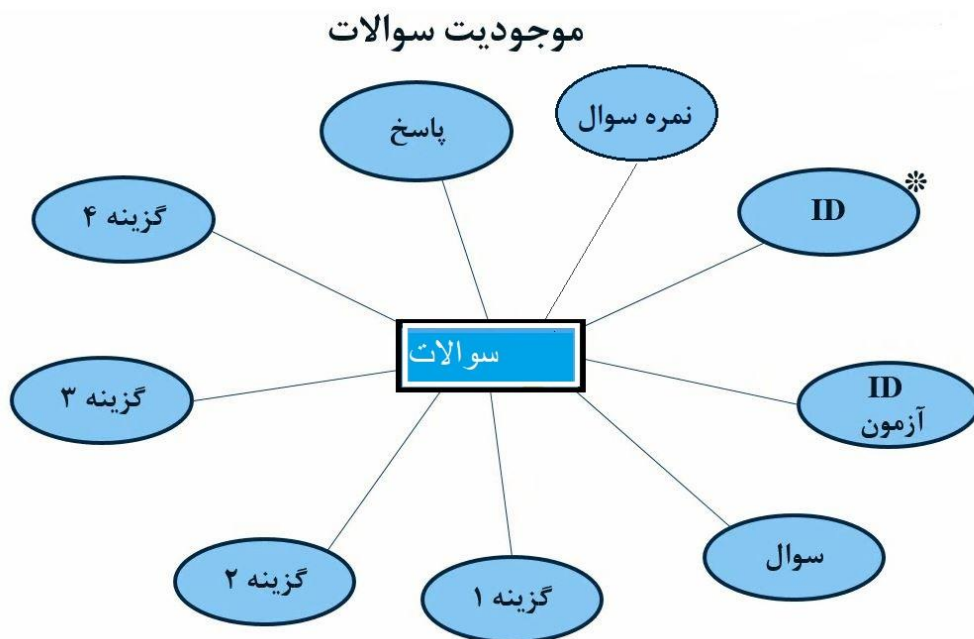
شکل (۹-۲) موجودیت آزمون



به طور کلی، موجودیت "آزمون" در یک سیستم مدیریت آزمون دارای ویژگی‌های مختلفی است که می‌توانند به صورت زیر توضیح داده شوند:

- **کد آزمون:** یک شناسه یا کد یکتا برای شناسایی هر آزمون در سیستم.
- **نام آزمون:** عنوان یا نام معیاری که برای آزمون تعریف شده است.
- **تعداد سوالات:** تعداد کل سوالاتی که در قالب این آزمون وجود دارد.
- **مدت زمان:** زمانی که برای انجام آزمون تعیین شده است.
- **تاریخ و زمان آزمون:** تاریخ و زمان مشخصی که برای برگزاری آزمون در نظر گرفته شده است.

## ۱-۲-۳-۱۰ موجودیت سوال



شکل (۱۰-۲) موجودیت سوالات

شناسه سوال: یک شناسه یا کد یکتا برای شناسایی هر سوال در سیستم.

متن سوال: متن یا محتوای سوال که ارائه می‌شود.

تصویر سوال: در صورت وجود، تصویر مرتبط با سوال.

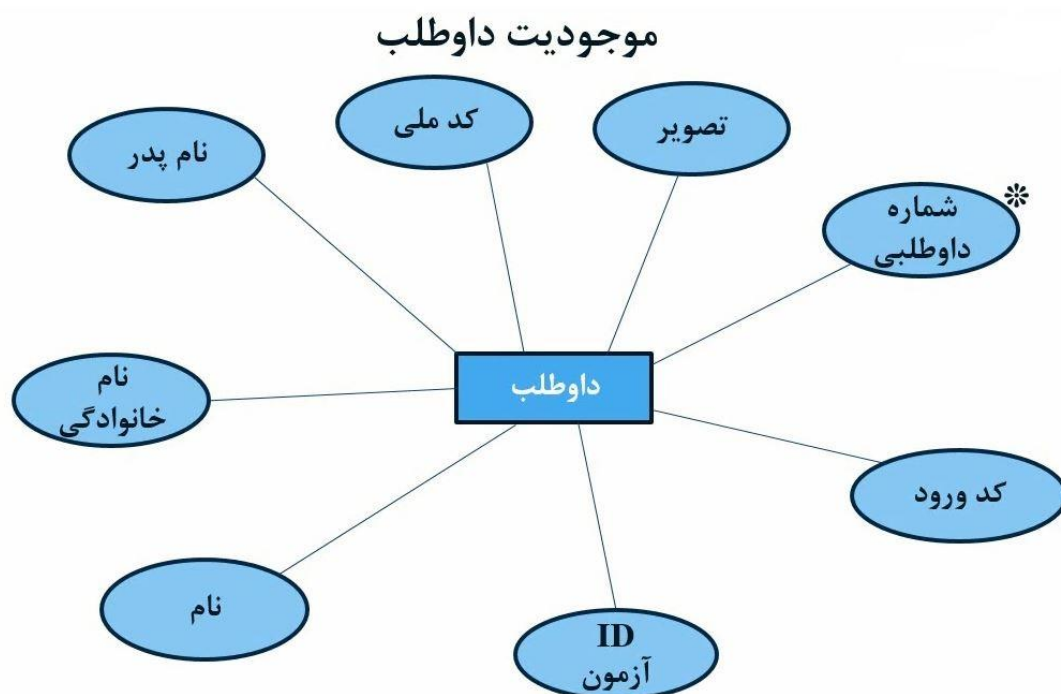
گزینه‌های پاسخ: گزینه‌های مختلفی که برای پاسخ به سوال ارائه شده‌اند.

تصاویر گزینه‌ها: در صورت وجود، تصاویر مرتبط با گزینه‌های سوال.

پاسخ صحیح: پاسخ صحیح به سوال.

وزن سوال: وزن یا اهمیتی که به سوال اختصاص داده شده است

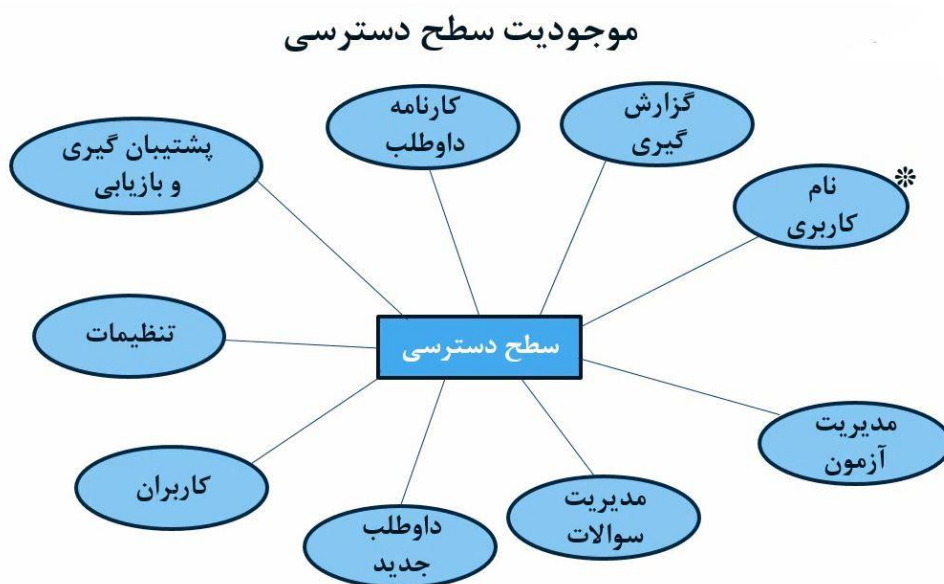
## ۲-۱-۳-۱۱ موجودیت داوطلب



شکل (۱۱-۲) موجودیت داوطلب

داوطلب در یک سیستم مدیریت آزمون معمولاً نشان‌دهنده فردی است که در آزمون شرکت می‌کند یا به عنوان کاندیدا در سیستم ثبت نام کرده است. ویژگی‌ها و وظایف موجودیت داوطلب در این سیستم ممکن است شامل موارد زیر باشد:

## ۲-۱-۳-۱۲ موجودیت سطح دسترسی



شکل (۱۲-۲) موجودیت سطح دسترسی

این موجودیت سطح دسترسی به نوعی گویا می‌تواند توسط مدیر سیستم برای هر کارمند به صورتی اختصاص یابد که این کارمند چه اجازه‌ها و دسترسی‌هایی را در سیستم دارد. این سطح دسترسی می‌تواند شامل مجوزها و قابلیت‌های مشخصی برای هر کاربر باشد. به طور کلی می‌توانید برای مدل‌سازی موجودیت "سطح دسترسی"، ویژگی‌هایی مانند زیر را در نظر بگیرید:

- \*\*مجوزها و دسترسی‌ها\*\*\*: لیستی از اجازه‌ها و دسترسی‌های مشخص برای هر سطح دسترسی.

- \*\*شناسه یا نام کاربری\*\*:: اختصاص یکتایی که به هر کاربر (کارمند) در سیستم داده می‌شود.

این موجودیت می‌تواند دارای روابطی با موجودیت‌های دیگر سیستم باشد، به عنوان مثال:

- \*\*سطح دسترسی (Access Level)\*\* <-- کارمند:: نشان دهنده دسترسی کارمندان به اجزای مختلف سیستم.

- \*\*سطح دسترسی (Access Level)\*\* <-- مدیر سیستم:: برای نمایش ارتباطی که این سطح دسترسی با مدیر سیستم دارد، مثلاً برای تعیین یا تغییر سطح دسترسی کاربران.

به این ترتیب، مدیر سیستم می‌تواند به هر کارمند یک سطح دسترسی خاص اختصاص دهد که به طور خاص تعیین کننده مجوزها و دسترسی‌های آن کارمند در سیستم به هر بخش سیستم است.

ویژگی‌ها:



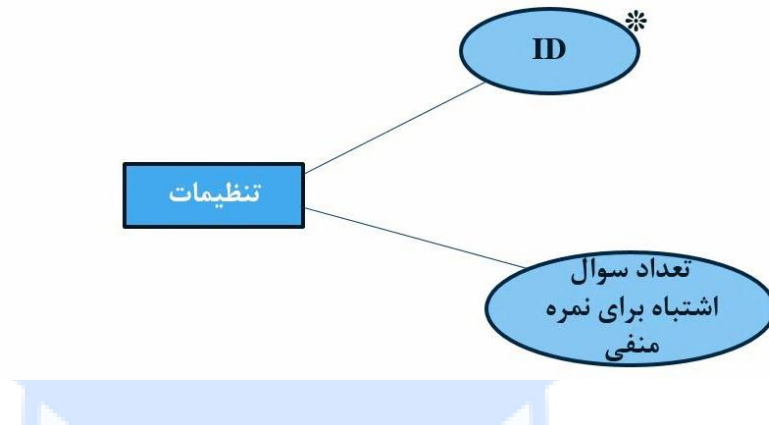
کد داوطلبی: یک شناسه یا کد یکتا که برای ورود به سیستم و شناسایی داوطلب استفاده می‌شود.

کد ورود: رمز مربوط به کد ورود برای احراز هویت در سیستم.

شناسه آزمون: شناسه کد آزمون که در آن نام نویسی شده است توسط مدیر سیستم یا یک کارمند.

مشخصات شخصی: اطلاعاتی مانند نام، نام خانوادگی، کد ملی، عکس پرسنلی و سایر اطلاعات هویتی.

## موجودیت تنظیمات



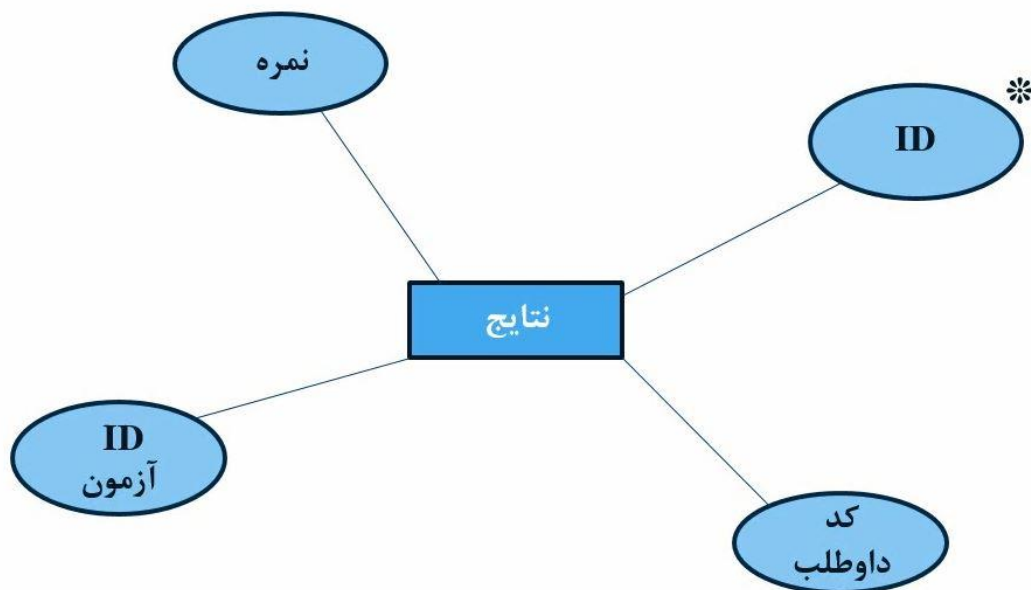
شکل (۲-۱۳) موجودیت تنظیمات

موجودیت "تنظیمات" برای مدل‌سازی ویژگی‌هایی که بر روی داوطلبان در آزمون تأثیر می‌گذارند این موجودیت می‌تواند ویژگی‌هایی مانند تعداد سوالاتی که نمره منفی را فعال می‌کند را نمایش دهد. این تنظیمات با موجودیت "آزمون" همبستگی داشته باشد و در صورتی که در آزمون این ویژگی‌هایی فعال شده باشند، بر روی نمره داوطلبان تأثیر گذار است.

بنابراین، ویژگی‌های زیر را برای موجودیت "تنظیمات" در نظر گرفته شده:

- \*\*شناسه (ID)\*\*: یک شناسه یکتا برای هر مورد تنظیمات.
  - \*\*تعداد سوالات با نمره منفی\*\*: تعداد سوالاتی که در صورت پاسخ نادرست منجر به کسر نمره می‌شوند.
- با تعریف روابط مناسب، این موجودیت می‌تواند با موجودیت "آزمون" همبستگی داشته باشد تا هنگام تنظیم آزمون، تأثیرات موردنظر بر روی داوطلبان را اعمال کند.

## موجودیت نتایج



شکل (۲-۱۴) موجودیت نتایج

موجودیت "نتایج" یا "Results" در واقع وضعیت و نتیجه‌ی هر داوطلب بعد از انجام آزمون را نمایش می‌دهد. این موجودیت وابسته به موجودیت "آزمون" و "داوطلب" است و ویژگی‌های مهمی می‌تواند شامل موارد زیر باشد:

**\*\* شناسه: (ID)\*\*** یک شناسه یکتا برای هر نتیجه‌ی آزمون.

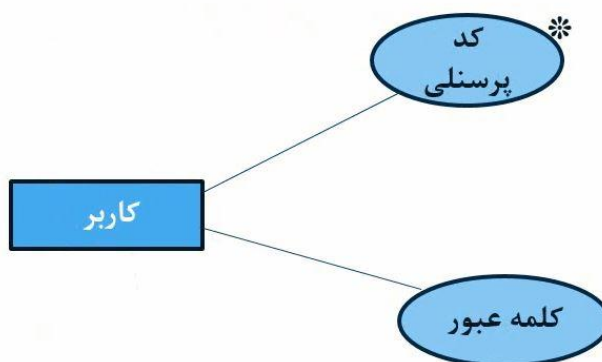
**\*\* شناسه آزمون: (Exam ID)\*\*** مشخصه‌ای که به هر آزمون نسبت داده می‌شود و این موجودیت را به آزمون مرتبط متصل می‌کند.

**\*\* کد داوطلب: (Candidate Code)\*\*** شناسه یکتای داوطلب که نتایج آزمون به آن مرتبط می‌شود.

**\*\* نمره: (Score)\*\*** نمره‌ی کسب شده توسط داوطلب در آزمون.

با این موجودیت می‌توان مدیریت نتایج هر آزمون را انجام داد و با استفاده از شناسه‌ها، ارتباطات موردنیاز بین داوطلبان و آزمون‌های مختلف را برقرار کرد. این نتایج می‌توانند به عنوان یک منبع ارزشمند برای ارزیابی عملکرد داوطلبان و تحلیل‌های مختلف در آزمون‌های مختلف استفاده شوند.

## موجودیت کاربر



شکل (۲-۱۵) موجودیت کاربر

موجودیت "کاربر" با دو ویژگی اصلی "کد پرسنلی" و "کلمه عبور" مشخص می‌شود. این موجودیت در سیستم نقشی ایفا می‌کند که از ورود و استفاده از قابلیت‌های سیستم برای انجام وظایف خودش استفاده می‌کند.

این دو ویژگی به کاربر اجازه می‌دهند که با وارد کردن کد پرسنلی و کلمه عبور خود، به سیستم وارد شوند و سپس از قابلیت‌ها و عملکردهای مختلف سیستم استفاده کنند. به طور معمول، کد پرسنلی به عنوان یک شناسه یکتا برای هر کاربر در نظر گرفته می‌شود و کلمه عبور نیز برای تأیید هویت و اعتبارسنجی کاربر استفاده می‌شود.

این ویژگی‌ها به کاربران اجازه می‌دهند تا با تأیید هویت خود و احراز هویت، از امکانات و خدمات سیستم استفاده کنند و عملکردهای مورد نیاز خود را انجام دهند.

## ب-۲ روابط موجودیت ها

برخی موجودیت‌ها در سیستم ما با یکدیگر ارتباط دارند :

- آزمون و سوالات:

- رابطه یک به چند بین آزمون و سوالات وجود دارد. هر آزمون می‌تواند دارای یک یا چند سوال باشد، اما هر سوال به طور تنها متعلق به یک آزمون است.

- داوطلب و نتایج:

- رابطه یک به چند بین داوطلب و نتایج وجود دارد. یک داوطلب می‌تواند در یک یا چند آزمون شرکت کند. هر نتیجه متعلق به یک داوطلب است، اما هر داوطلب ممکن است چندین نتیجه داشته باشد.

- کاربر و سطح دسترسی:

- رابطه یک به یک بین کاربر و سطح دسترسی وجود دارد. به این معنا که هر کاربر مرتبط با یک سطح دسترسی مشخص است و هر سطح دسترسی نیز متعلق به یک کاربر است.

این ارتباطات مشخص می‌کنند چگونه اطلاعات در سیستم شما می‌توانند به یکدیگر وابسته باشند و چگونه موجودیت‌ها با یکدیگر در ساختار سیستم تعامل دارند. طراحی پایگاه داده

## ب-۳ طراحی پایگاه داده

در این مرحله اول ، با استفاده از (SQL Server Management Studio (SSMS ، یک پایگاه داده جدید با نام dbazmoon را ایجاد کرده. این پایگاه داده به عنوان مخزن اصلی اطلاعات برای سیستم آزمون گیری عمل می‌کند. سپس با استفاده از موجودیت هایی که برای سیستم بررسی کرده ایم و روابط و ویژگی های آن جداولی در پایگاه داده خود ایجاد میکنیم .



## ۱۶-۳-۱-۲ جداول پایگاه داده

	Column Name	Data Type	Allow Nulls
🔑	UserName	nvarchar(50)	<input type="checkbox"/>
	Azmoon	bit	<input checked="" type="checkbox"/>
	Questions	bit	<input checked="" type="checkbox"/>
	Students	bit	<input checked="" type="checkbox"/>
	Users	bit	<input checked="" type="checkbox"/>
	Setting	bit	<input checked="" type="checkbox"/>
	BackupRestore	bit	<input checked="" type="checkbox"/>
	Workbook	bit	<input checked="" type="checkbox"/>
	Report	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

شکل (۱۶-۲) جدول سطح دسترسی

	Column Name	Data Type	Allow Nulls
🔑	ID	int	<input type="checkbox"/>
	TestName	nvarchar(50)	<input checked="" type="checkbox"/>
	Testdate	nchar(10)	<input checked="" type="checkbox"/>
	Timeout	int	<input checked="" type="checkbox"/>
	NegetiveScore	bit	<input checked="" type="checkbox"/>
	AcceptableScore	int	<input checked="" type="checkbox"/>
	QuestionCount	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

شکل (۱۷-۲) جدول آزمون

اداسگاه اراداسلا  
واحد نجف آباد

	Column Name	Data Type	Allow Nulls
▶	ID	int	<input type="checkbox"/>
	TestID	int	<input checked="" type="checkbox"/>
	Question	nvarchar(MAX)	<input checked="" type="checkbox"/>
	QuestionIMG	varbinary(MAX)	<input checked="" type="checkbox"/>
	option1	nvarchar(MAX)	<input checked="" type="checkbox"/>
	option2	nvarchar(MAX)	<input checked="" type="checkbox"/>
	option3	nvarchar(MAX)	<input checked="" type="checkbox"/>
	option4	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Answer	nvarchar(MAX)	<input checked="" type="checkbox"/>
	option1IMG	varbinary(MAX)	<input checked="" type="checkbox"/>
	option2IMG	varbinary(MAX)	<input checked="" type="checkbox"/>
	option3IMG	varbinary(MAX)	<input checked="" type="checkbox"/>
	option4IMG	varbinary(MAX)	<input checked="" type="checkbox"/>
	Qscore	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

شكل (٢-١٨) جدول سوالات

	Column Name	Data Type	Allow Nulls
▶	ID	int	<input type="checkbox"/>
	StudentNumber	int	<input checked="" type="checkbox"/>
	Score	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

شكل (٢-١٩) جدول نتائج

	Column Name	Data Type	Allow Nulls
▶	ID	int	<input type="checkbox"/>
	NegativeScoreCount	int	<input checked="" type="checkbox"/>

شكل (٢-٢٠) جدول تنظيمات

	Column Name	Data Type	Allow Nulls
▶	StudentNumber	int	<input type="checkbox"/>
	LoginCode	nvarchar(50)	<input checked="" type="checkbox"/>
	TestID	int	<input checked="" type="checkbox"/>
	FirsName	nvarchar(50)	<input checked="" type="checkbox"/>
	LastName	nvarchar(50)	<input checked="" type="checkbox"/>
	FatherName	nvarchar(50)	<input checked="" type="checkbox"/>
	CodeMelli	nchar(10)	<input checked="" type="checkbox"/>
	Picture	varbinary(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

شکل (۲۱-۲) جدول داوطلبان

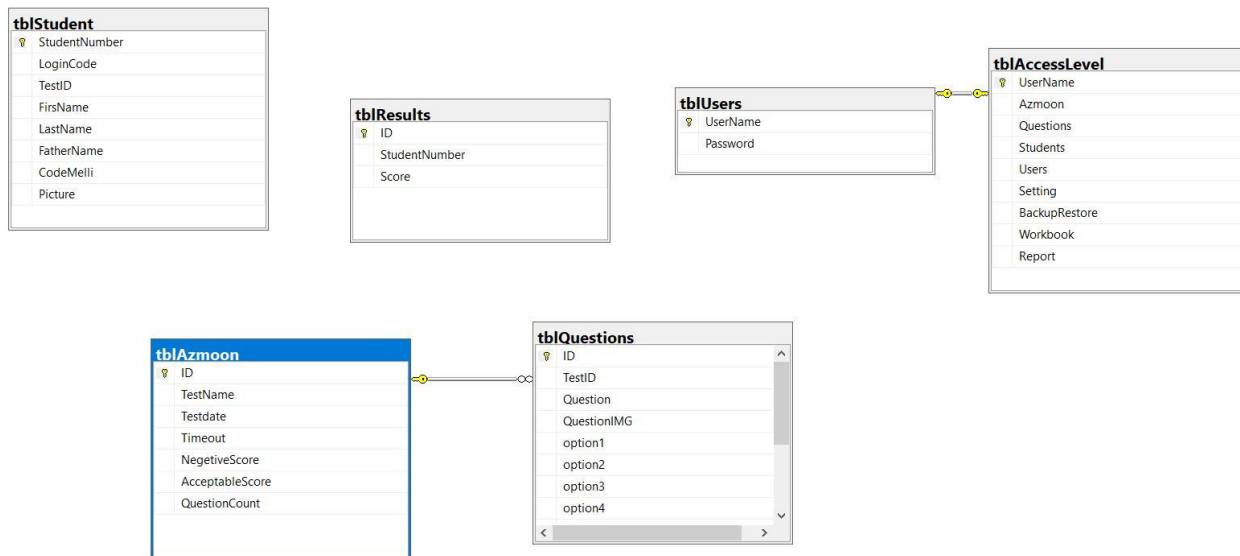
	Column Name	Data Type	Allow Nulls
▶	UserName	nvarchar(50)	<input type="checkbox"/>
	Password	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

شکل (۲۲-۲) جدول کاربر



## ۲-۱-۳ روابط جداول پایگاه داده

پیاده سازی روابط موجودیت ها در مدل داده ای یا دیاگرام برای جداول پایگاه داده:



شکل (۲-۳) مدل داده ای یا دیاگرام جداول پایگاه داده

## ۲-۱-۴ پیاده سازی محیط کاربری

محیط پیاده سازی برنامه ما در برنامه visual studio انجام شده است که با استفاده از زبان برنامه نویسی c# و محیط های windows form application ها انجام می شود.



با توجه به نمودار ها و رفتار و ویژگی use case ها محیط های گرافیکی (کاربری) بصورت زیر تقسیم بندی و طراحی می شود:

## ۲-۱-۴ فرم اصلی

شامل تمام ویژگی های طراحی شده سطح دسترسی برای مدیر و کارمندان سیستم با استفاده از دکمه هایی نقش مبانبر برای دسترسی به فرم های نام برده را دارند



شکل (۲-۲۴) فرم اصلی

## ۲-۱-۴ فرم ورود

اولین فرم که پس از اجرای برنامه اجرا میشود و داوطلبان یا مدیر یا کارمند به برنامه دسترسی پیدا میکنند در اولین اجرا برنامه فرم ورود بصورت زیر نمایش داده خواهد شد زیرا اولین ورود به سیستم محسوب میشود پس فضای نمایشی بصورتی است که کاربر (مدیر) رمز و پسورد دلخواه خویش که دسترسی نامحدود دارد وارد کند

واحد نجف آباد

شکل (۲۵-۲) فرم ورود-۱

در اجراهای بعد فرم دکمه ثبت حذف و دکمه انصراف قرار میگیرد و اگر هر یک از radiobutton های کارمند یا داوطلب انتخاب شود نام label ها به اینصورت تعویض میشود

شکل (۲۶-۲) فرم ورود-۲

## ۲-۱-۴ فرم مدیریت آزمون

این فرم جهت وارد کردن اطلاعاتی که باید در جدول آزمون ذخیره شود و حذف و ویرایش اطلاعات وارد شده است و همچنین در پایین فرم یک datagridview قرار گرفته است که به اطلاعات جدول قابل ویرایش و نمایش باشد.

واحد بسپار



این فرم جهت وارد کردن سوال هر آزمون که انتخاب شده است و جواب سوال که باید در جدول سوالات ذخیره شود و امکان حذف و ویرایش اطلاعات وارد شده است و همچنین در پایین فرم یک datagridview قرار گرفته است که تمام سوالات آزمون انتخابی در جدول دیده خواهد شد قابل ویرایش باشد .

پس از وارد کردن سوالات آزمون کاربر وارد صفحه ارزش گذاری سوالات جهت تغییر بار یا وزن یا نمره هر سوال می باشد که شامل یک datagrid view که هر سوال ارزش مشخص خود را دارد مشخصی را دارد .

شماره سوال	صورت سوال	ارزش سوال
	buttonItem1	

شکل (۲-۲۹) فرم مدیریت سوالات-۲



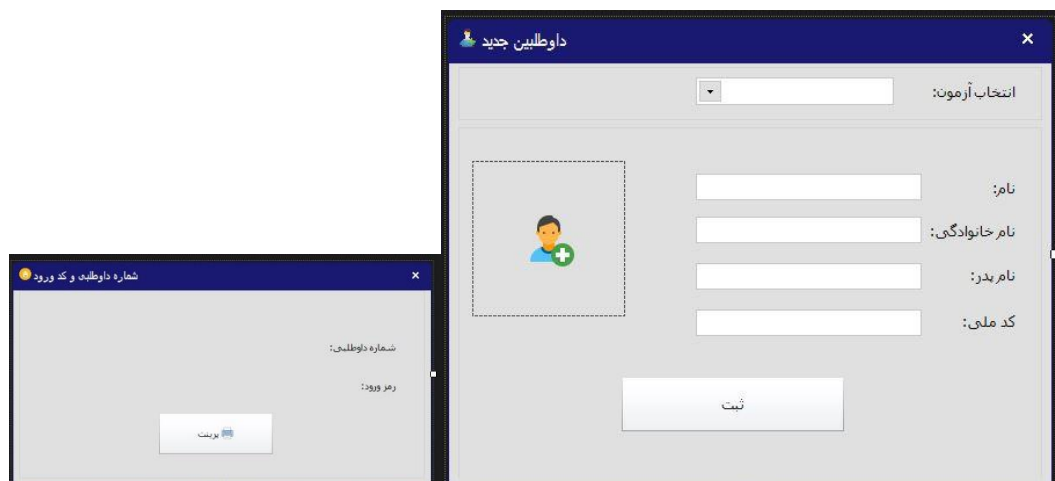
## ۴-۱-۲ فرم گزارش

جهت تهیه فایل گزارش باتوجه به فیلتر انتخاب شده از فرم



شکل (۳۰-۲) فرم گزارش

## ۴-۱-۲-۵ فرم داوطلب



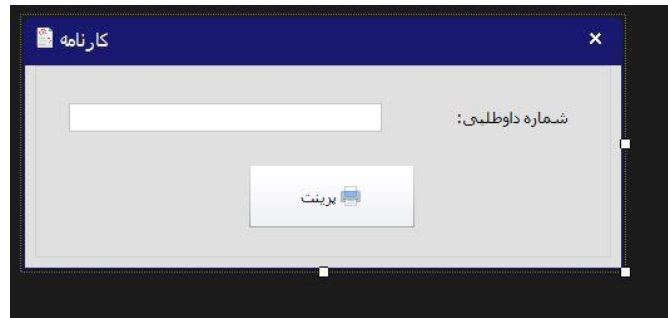
شکل (۳۱-۲) فرم داوطلب

وارد کردن اطلاعات داوطلب باتوجه به آزمون انتخاب شده.

پس از ثبت نمایش فرم چاپ کد داوطلبی و کد ورود داوطلب

## ۲-۱-۴-۶ فرم کارنامه

نمایش و چاپ نتیجه آزمون داوطلبی که شماره داوطلبی آن وارد شده.



شکل (۲-۳۲) فرم کارنامه

## ۲-۱-۴-۶ فرم تنظیمات

وارد کردن تعداد سوالات باعث یک نمره منفی برای آزمون میشود



شکل (۲-۳۳) فرم تنظیمات

## ۲-۲ پیاده سازی

### ۲-۲-۱ پیاده سازی بخش اصلی

#### ایجاد فرم

پس از ایجاد پروژه جدید، یک فرم جدید در محیط Visual Studio باز می‌شود. این فرم، پایه و اساس Windows Form شما خواهد بود.

#### اضافه کردن دکمه‌ها

برای اضافه کردن دکمه‌ها به فرم، از منوی Toolbox گزینه Button را انتخاب کنید و سپس دکمه را در محل مورد نظر خود در فرم بکشید.

#### اضافه کردن کد

برای اضافه کردن کد به دکمه‌ها، روی دکمه کلیک راست کرده و گزینه View Code را انتخاب کنید. سپس، کد مورد نظر خود را در پنجره کدنویسی وارد کنید.

#### تست فرم

برای تست فرم، از منوی Debug گزینه Start را انتخاب کنید. فرم در پنجره Design باز می‌شود. می‌توانید با کلیک کردن روی دکمه‌ها، عملکرد آنها را تست کنید.

پیاده سازی Windows Form در این پروژه

در این پروژه، یک Windows Form با هشت دکمه ایجاد شده است. هر دکمه یک عملکرد خاص را انجام می‌دهد. مراحل پیاده سازی این Windows Form به شرح زیر است:

#### ایجاد پروژه جدید

در ابتدا، یک پروژه جدید با نام Azmoon ایجاد می‌کنیم. نوع پروژه Windows Forms Application است. اضافه کردن فرم

پس از ایجاد پروژه جدید، یک فرم جدید با نام frmmain در محیط Visual Studio باز می‌شود.

#### اضافه کردن دکمه‌ها

از منوی Toolbox گزینه Button را انتخاب می‌کنیم و سپس دکمه‌ها را در محل مورد نظر خود در فرم می‌کشیم. اضافه کردن کد

برای هر دکمه، کدی را به شرح زیر اضافه می‌کنیم:

```
#C
private void btnAzmoonM_Click(object sender, EventArgs e)
{
    frmAzmoon Azmoon = new frmAzmoon();
    Azmoon.Show();
}
```

این کد، یک فرم جدید با نام frmAzmoon باز می‌کند.

برای سایر دکمه‌ها نیز کدهای مشابهی را اضافه می‌کنیم.

#### توضیحات تکمیلی

برای مثال، می‌توان از ویژگی BackColor برای تغییر رنگ پس‌زمینه دکمه استفاده کرد.

در این پروژه، از ویژگی AccessLevel برای فعال یا غیرفعال کردن دکمه‌ها بر اساس سطح دسترسی کاربر استفاده شده است. برای مثال، دکمه btnAzmoonM فقط برای کاربرانی که سطح دسترسی خاصی دارند، فعال خواهد بود.

## ۲-۲-۲ پیاده سازی بخش ورود

در اینجا یک توضیح کامل از پیاده سازی فرم Windows ارائه شده است:

ساختار کلی

فرم Windows به سه بخش اصلی تقسیم شده است:

- کنترل های ورود به سیستم: این بخش شامل کنترل های ورود به سیستم، مانند نام کاربری، گذرواژه و دکمه رادیویی برای انتخاب اینکه کاربر کارمند یا کاندیدا است.
- کنترل های ورود به سیستم
- کنترل های ورود به سیستم عبارتند از:
- RadioButtonEmp: این دکمه رادیویی به طور پیش فرض انتخاب شده است. وقتی انتخاب شود، برچسب و کادر متن زیر آن به "نام کاربری" و "کلمه عبور" تغییر نام می دهند.
- RadioButtonVal: این دکمه رادیویی برای ورود به سیستم به عنوان یک کاندیدا استفاده می شود. وقتی انتخاب شود، برچسب و کادر متن زیر آن به "شماره داوطلب" و "کد ورود" تغییر نام می دهند.
- textUser: این کادر متن برای ورود نام کاربری یا شماره دانشجویی کاربر استفاده می شود.
- textPass: این کادر متن برای ورود گذرواژه یا کد ورود کاربر استفاده می شود.
- btnLogin: این دکمه برای ورود به سیستم استفاده می شود.

دکمه ها

دکمه ها عبارتند از:

- btnCancel: این دکمه برای لغو تلاش ورود به سیستم و بستن فرم استفاده می شود.
- btnAddadmin: این دکمه فقط در صورتی قابل مشاهده است که کاربر موجودی در پایگاه داده وجود نداشته باشد. برای افزودن مدیر جدید استفاده می شود.

PanelEx1

این پانل در ابتدا پنهان است. حاوی موارد زیر است:

- btnAddadmin: این دکمه برای افزودن مدیر جدید استفاده می شود.

دسترسی به داده ها

فرم Windows از پایگاه داده ای به نام DBazmon برای ذخیره اطلاعات کاربر استفاده می کند. این پایگاه داده شامل جدول های زیر است:

- tblUsers: این جدول اطلاعات کاربر را ذخیره می کند، از جمله نام کاربری، گذرواژه و سطح دسترسی.
- tblAccessLevels: این جدول سطح دسترسی هر کاربر را ذخیره می کند.

تعامل کاربر

فرم Windows موارد زیر را مدیریت می کند:

- `RadioButtonEmp_CheckedChanged`: این رویداد زمانی رخ می دهد که دکمه رادیویی `RadioButtonEmp` یا `RadioButtonVal` انتخاب می شود. برجسب ها و کادرهای متن را به ترتیب به روز می کند.
- `btnLogin_Click`: این رویداد زمانی رخ می دهد که دکمه `btnLogin` کلیک شود. سعی می کند کاربر را وارد سیستم کند و اگر موفقیت آمیز باشد، فرم اصلی برنامه را باز می کند.
- `textUser_KeyPress`: این رویداد زمانی رخ می دهد که کادر متن `textUser` فشار داده شود. اگر دکمه رادیویی `RadioButtonVal` انتخاب شده باشد، از کاربر جلوگیری می کند که کاراکترهای غیر عددی وارد کند.
- `btnCancel_Click`: این رویداد زمانی رخ می دهد که دکمه `btnCancel` کلیک شود. فرم را می بندد.
- `btnAddadmin_Click`: این رویداد زمانی رخ می دهد که دکمه `btnAddadmin` کلیک شود. یک مدیر جدید به پایگاه داده اضافه می کند.

ویژگی های بیشتر

فرم Windows همچنین شامل ویژگی های زیر است:

- رمزنگاری AES: رمز عبور قبل از ذخیره در پایگاه داده رمزگذاری می شود.
- اعتبار سنجی کاربر: نام کاربری و گذرواژه قبل از ورود کاربر به سیستم اعتبارسنجی می شوند.
- بررسی سطح دسترسی: سطح دسترسی کاربر قبل از اینکه اجازه دسترسی به برخی ویژگی های برنامه را داشته باشد بررسی می شود.
- مدیریت خطا: فرم Windows خطاها را به طور محترمانه مدیریت می کند و پیام های مناسب را به کاربر نمایش می دهد.
- دکمه ها: این بخش شامل دکمه های ورود به سیستم و افزودن مدیر جدید می باشد.
- `PanelEx1`: این پنل در ابتدا پنهان است و فقط در صورت وجود کاربر موجود در پایگاه داده نمایش داده می شود. حاوی یک دکمه برای افزودن مدیر جدید است.

۲-۳ پیاده سازی فرم مدیریت آزمون

واحد نجف آباد

این کد یک فرم ویندوزی به نام "frmAzmoon" را پیاده‌سازی می‌کند که برای مدیریت آزمون‌ها استفاده می‌شود. این فرم از کلاس "Office2007Form" ارث بری کرده و از فضای نام‌های مختلفی مانند "System", "DevComponents.DotNetBar" و "System.Data.Entity" استفاده می‌کند.

در این فرم، یک شیء از کلاس "DBazmonEntities" به نام "db" ایجاد شده است که برای دسترسی به داده‌ها از پایگاه داده استفاده می‌شود.

در رویداد Load فرم، یک BindingSource به نام "bs" ایجاد شده و اطلاعات جدول "tblAzmoons" از پایگاه داده بارگذاری شده و به آن متصل می‌شود. سپس DataGridView به نام "Dgrid" به این BindingSource متصل می‌شود تا اطلاعات نمایش داده شود. همچنین در این رویداد، تعداد ستون‌های نمایش داده شده بررسی شده و در صورتی که بیشتر از ۷ ستون باشد، تغییراتی اعمال می‌شود.

در رویداد Click دکمه "buttonX1"، ابتدا بررسی می‌شود که آیا نام آزمون تکراری است یا خیر. سپس بسته به نوع عملیات (جدید یا ویرایش)، اطلاعات آزمون جدید ثبت می‌شود یا اطلاعات آزمون موجود ویرایش می‌شود.

در رویداد CellFormatting اطلاعات ستون‌های خاصی از جدول تغییر داده می‌شود.

در رویداد Click دکمه‌های "buttonItem2" و "buttonItem3" نیز عملیات حذف و ویرایش بر روی آزمون‌ها انجام می‌شود.

در کل، این فرم برای مدیریت آزمون‌ها و انجام عملیات افزودن، ویرایش و حذف آن‌ها ایجاد شده است.

## ۴-۲-۲ پیاده‌سازی فرم مدیریت سوالات

این کد یک فرم Windows است که یک واسط برای مدیریت سوالات یک آزمون فراهم می‌کند. این فرم شامل چند قسمت مختلف است که امکاناتی مثل افزودن، ویرایش، و حذف سوالات را فراهم می‌کند. همچنین تصاویر مربوط به سوالات و گزینه‌ها را هم می‌توانید به آن اضافه کنید. این کد از اتصال به پایگاه داده استفاده می‌کند تا سوالات را از آن خوانده و در آن ذخیره کند. این یک برنامه کامل و پیچیده است و برای توضیح دادن مراحل کامل پیاده‌سازی آن، نیاز به محیط برنامه نویسی خاص و پروژه در دسترس نیاز داریم.

بهتر است مراحل پیاده‌سازی این فرم را به چندین بخش تقسیم کنیم:

۱. **استفاده از کتابخانه‌ها و ایجاد فرم**: در این قسمت، کتابخانه‌های مورد نیاز برای کار با فرم و اجزای آن فراخوانی شده و یک فرم جدید ساخته شده است.
  ۲. **اتصال به پایگاه داده**: این بخش شامل ایجاد اتصال به پایگاه داده و لود کردن اطلاعات مورد نیاز برای نمایش در فرم است.
  ۳. **توابع برای عملیات افزودن و ویرایش و حذف سوالات**: این بخش شامل توابعی است که وظیفه افزودن، ویرایش و حذف سوالات را دارند. این عملیات با استفاده از دکمه‌ها و کلیک روی المان‌های موجود در فرم صورت می‌گیرد.
  ۴. **توابع مربوط به عکس‌ها**: این قسمت شامل توابعی است که تصاویر را به آرایه‌های بایت و برعکس تبدیل می‌کنند.
  ۵. **رویدادهای مربوط به المان‌های فرم**: این بخش شامل رویدادهای مربوط به المان‌های فرم مانند کلیک روی دکمه‌ها، تغییرات در انتخاب‌ها و ورودی‌ها، و رویدادهای مربوط به درگ و دراپ عکس‌ها است.
- این کد بر اساس یک پروژه و پایگاه داده خاص نوشته شده است. توجه داشته باشید که برای اجرای این کد، می‌بایستی تنظیمات و ارتباط با یک پایگاه داده موجود در پروژه مورد نظر را داشته باشید.

## ۲-۲-۵ پیاده‌سازی فرم مدیریت سوالات-۲

این کد یک فرم Windows است که به کمک پایگاه داده‌ی خود (موجود در کلاس `DBazmonEntities`)، اطلاعات مربوط به سوالات آزمون‌ها را نمایش می‌دهد و اجازه می‌دهد تا اطلاعات را ویرایش کنید. دارای عناصر متفاوتی است که از آنها برای نمایش اطلاعات و انجام عملیات استفاده می‌کند.

مراحل پیاده‌سازی این فرم به شرح زیر است:

واحد محف اباد

۱. \*\*\*تعریف فرم و متغیرها\*\*\*: در این بخش، یک فرم جدید ایجاد شده است که از کتابخانه‌ها و پایگاه داده استفاده می‌کند. همچنین یک متغیر برای مدیریت بایندینگ داده‌ها (`BindingSource`) و دو فرم دیگر (`frmQuestion`) نیز تعریف شده‌اند.

۲. \*\*\*لود داده‌ها و نمایش آنها در فرم\*\*\*: در رویداد `Load` فرم (`Form1\_Load`)، داده‌های مربوط به آزمون‌ها از پایگاه داده خوانده شده و در کمبوباکس (`comboBox1`) نمایش داده می‌شوند. همچنین سوالات مربوط به آزمون انتخاب شده به وسیله `BindingSource` بارگذاری و در دیتاگراید (`DataGridView1`) نمایش داده می‌شوند.

۳. \*\*\*ویرایش اطلاعات سوالات\*\*\*: وقتی کاربر روی دکمه‌ی ویرایش کلیک می‌کند (`buttonItem2\_Click\_2`)، اطلاعات سوال انتخابی در دیتاگراید نمایش داده می‌شوند و کاربر می‌تواند اطلاعات مربوط به امتیاز سوال را ویرایش کند. این اطلاعات شامل میزان امتیاز مورد نیاز برای هر سوال است.

۴. \*\*\*تغییرات در سوالات و ذخیره‌سازی\*\*\*: وقتی کاربر تغییری در میزان امتیاز سوالات ایجاد می‌کند و روی دکمه‌ی ذخیره کلیک می‌کند (`buttonX1\_Click`)، این تغییرات به پایگاه داده ذخیره می‌شوند.

۵. \*\*\*نمایش سوالات بر اساس انتخاب کاربر\*\*\*: هنگامی که کاربر یک آزمون را از کمبوباکس انتخاب می‌کند (`comboBox1\_SelectedIndexChanged`)، سوالات مربوط به آزمون انتخابی در دیتاگراید نمایش داده می‌شوند.

این کد بر اساس پروژه‌ای خاص و پایگاه داده‌ای خاص نوشته شده است و برای اجرای صحیح آن، لازم است که تنظیمات و ارتباط با پایگاه داده مورد نظر در پروژه انجام شده باشد.

## ۲-۶ پیاده سازی فرم گزارش

این کد یک فرم Windows است که از Stimulsoft Reports برای ایجاد گزارشات PDF استفاده می‌کند. این فرم اجازه می‌دهد تا گزارشات مختلفی مانند لیست آزمون‌ها، سوالات آزمون، داوطلبان گذرانده و غیره را ایجاد و نمایش دهد.

• مراحل پیاده‌سازی:



۱. **تعریف فرم و متغیرها**: در این بخش، یک فرم جدید ایجاد شده است که از کتابخانه‌ها و کلاس‌های مختلف استفاده می‌کند. همچنین یک ارتباط با پایگاه داده ('DBazmonEntities') و یک فرم دیگر نیز تعریف شده‌اند.

۲. **لود داده‌ها و تنظیمات اولیه**: در رویداد 'Load' فرم ('frmReport\_Load')، داده‌های مربوط به آزمون‌ها از پایگاه داده خوانده شده و در کمبوباکس ('comboBoxAZ') نمایش داده می‌شوند.

۳. **ایجاد و چاپ گزارشات PDF**:

- ایجاد گزارش بر اساس تاریخ آزمون ('radioOneDate'): با استفاده از iTextSharp، اطلاعات آزمون‌هایی که در تاریخ خاصی برگزار شده‌اند را به صورت یک جدول در یک فایل PDF ذخیره می‌کند.

- ایجاد گزارش بر اساس سوالات آزمون ('radioQuestions'): بر اساس آزمونی که کاربر انتخاب کرده است، اطلاعات مربوط به سوالات آن آزمون را به صورت جدول در یک فایل PDF ذخیره می‌کند.

- ایجاد گزارش بر اساس داوطلبانی که قبول شده‌اند ('radioPass'): اطلاعات داوطلبانی که نمره آن‌ها برای یک آزمون خاص به مرز قابل قبولی رسیده است، به صورت یک جدول در یک فایل PDF ذخیره می‌شود.

- ایجاد گزارش بر اساس همه داوطلبان یک آزمون ('radioButton4'): اطلاعات تمام داوطلبان یک آزمون خاص را به صورت یک جدول در یک فایل PDF ذخیره می‌کند.

این کد بر اساس پروژه‌ای خاص و ارتباط با یک پایگاه داده خاص نوشته شده است. برای اجرای صحیح آن، لازم است که تنظیمات و ارتباط با پایگاه داده مورد نظر در پروژه انجام شده باشد و فایل‌های مورد نیاز برای ایجاد گزارشات (مانند فونت‌ها و تصاویر) در مسیرهای معین موجود باشند.

## ۲-۷ پیاده سازی فرم داوطلب

کد فرم 'frmNewVal' یک فرم ویندوز است که به کاربر امکان می‌دهد برای یک آزمون جدید، داوطلبان جدید ثبت کند و اطلاعات آن‌ها را وارد کند.

مراحل پیاده‌سازی:

۱. **تعریف فرم و متغیرها**: ابتدا یک فرم جدید ایجاد شده است که از کتابخانه‌ها و کلاس‌های مختلف استفاده می‌کند. همچنین ارتباط با پایگاه داده و تعریف متغیرهای مورد نیاز انجام شده است.

۲. **لود داده‌ها و تنظیمات اولیه**: در رویداد 'Load' فرم ('frmNewVal\_Load')، داده‌های مربوط به آزمون‌ها از پایگاه داده خوانده شده و در کمبوباکس ('comboSelectAzNewVal') نمایش داده می‌شوند.

۳. **انتخاب تصویر**: هنگام کلیک روی 'pictureBoxVal'، یک پنجره باز می‌شود تا کاربر بتواند یک تصویر JPEG انتخاب کند و آن را برای داوطلب انتخاب کند.

۴. **ثبت داوطلب جدید**: هنگام کلیک بر روی دکمه 'btnPassNewVal'، ابتدا چک می‌شود که آیا داوطلب قبلاً در آزمون ثبت نام کرده یا خیر. سپس، اگر ثبت نام نکرده بود، یک کد تصادفی برای ورود داوطلب به سامانه تولید شده و اطلاعات داوطلب (شامل نام، نام خانوادگی، نام پدر، کد ملی، کد ورود، تصویر و آیدی آزمون) در جدول 'tblStudent' ذخیره می‌شود.

۵. **پاک کردن و بازنشانی فرم**: پس از ثبت داوطلب، فیلدهای ورودی پاک می‌شوند و تصویر به حالت پیش فرض باز می‌گردد.

۶. **نمایش کد ورود به داوطلب**: بعد از ثبت نام داوطلب، یک فرم دیگر به نام 'valencode' باز می‌شود که شامل کد ورود به داوطلب برای ورود به آزمون است.

این کد بر اساس پروژه‌های خاص و ارتباط با یک پایگاه داده خاص نوشته شده است. برای اجرای صحیح آن، لازم است که تنظیمات و ارتباط با پایگاه داده مورد نظر در پروژه انجام شده باشد و فرم‌ها و کلاس‌های مورد نیاز برای اجرای کد وجود داشته باشند.

## ۲-۲-۸ پیاده سازی فرم داوطلب - ۲

کد فرم 'valencode' یک فرم ویندوز است که کد ورود داوطلب به آزمون را نمایش می‌دهد و امکان ایجاد فایل PDF حاوی این اطلاعات را فراهم می‌کند.

مراحل پیاده‌سازی:

اداسگاه اراداسلا  
واحد نجف آباد

۱. **\*\*تعریف فرم و متغیرها\*\***: یک فرم ویندوز با عنوان `valencode` ایجاد شده است که در آن متغیرها و توابع مورد نیاز تعریف شده‌اند.

۲. **\*\*نمایش اطلاعات داوطلب\*\***: متد `GetStudentNumber` تعریف شده است تا اطلاعات مربوط به داوطلب (شامل شماره داوطلبی، کد ورود و نام آزمون) را نمایش دهد.

۳. **\*\*ساخت فایل PDF\*\***: در رویداد کلیک بر روی دکمه `buttonX1`، یک فایل PDF ایجاد می‌شود که شامل اطلاعات داوطلب (شماره داوطلبی و کد ورود) و نام آزمون است. این اطلاعات در یک جدول مرتب شده و در فایل PDF ذخیره می‌شوند.

۴. **\*\*نمایش فایل PDF\*\***: پس از ایجاد فایل PDF، آن را با استفاده از `System.Diagnostics.Process.Start` نمایش می‌دهد.

۵. **\*\*کامنت‌های خارج از استفاده\*\***: برخی از بخش‌های کد ممکن است کامنت شده باشند که احتمالاً برای استفاده‌های دیگری طراحی شده‌اند و در حال حاضر استفاده نمی‌شوند.

۶. **\*\*کد HTML (غیرفعال)\*\***: بخشی از کد از نوع HTML نیز برای ایجاد یک فایل HTML کامنت شده است، اما فعال نشده است.

۷. **\*\*استفاده از iTextSharp\*\***: کتابخانه iTextSharp برای ساخت فایل PDF استفاده شده است که می‌تواند نیازمندی‌های خاصی به تنظیمات مربوط به محیط محل اجرا داشته باشد.

این کد برای تولید یک فرم ویندوز با قابلیت ایجاد فایل PDF و نمایش کد دسترسی به آزمون برای داوطلب طراحی شده است. دقت کنید که برای اجرای صحیح آن، نیاز به محیط اجرایی و تنظیمات محیطی خاصی ممکن است داشته باشید.

## ۲-۱۰ پیاده سازی فرم نتایج

این قطعه کد یک فرم ویندوز به نام "result" است که نمرات یک دانشجو در آزمون را نمایش می‌دهد و قابلیت چاپ نمره‌ها به صورت فایل PDF را فراهم می‌کند.

مراحل پیاده‌سازی:

۱. **تعریف فرم و متغیرها**: یک فرم ویندوز با عنوان 'resault' ایجاد شده است که در آن متغیرها و توابع مورد نیاز تعریف شده‌اند.

۲. **نمایش نمرات دانشجو**: در رویداد کلیک بر روی دکمه 'btnPintRes'، نمرات یک دانشجو در آزمون خوانده می‌شوند. این اطلاعات شامل شماره دانشجویی، نام، نام خانوادگی، نام آزمون، نمره، و وضعیت (قبول یا مردود) است.

۳. **ساخت فایل PDF**: یک فایل PDF با نمرات دانشجو ایجاد می‌شود. از کتابخانه iTextSharp برای ساخت فایل PDF استفاده شده است. این فایل شامل یک جدول است که در آن اطلاعات نمرات نشان داده می‌شود.

۴. **نمایش فایل PDF**: پس از ساخت فایل PDF، با استفاده از 'System.Diagnostics.Process.Start' فایل PDF نمایش داده می‌شود تا قابلیت چاپ نمرات دانشجو فراهم شود.

این قطعه کد برای نمایش نمرات یک دانشجو و ایجاد یک فایل PDF برای آن طراحی شده است. لازم به ذکر است که برای اجرای صحیح کد، باید تنظیمات محیطی و مجوزهای لازم را داشته باشید.

## ۲-۲-۱۱ پیاده سازی فرم محیط آزمون

این برنامه ویندوزی با نام "frmExam" یک آزمون تعاملی را به کاربر ارائه می‌دهد و امکان پاسخ به سوالات، نمایش سوالات، و ارزیابی نمره را فراهم می‌کند.

مراحل پیاده‌سازی:

۱. **فرم و توابع اصلی**: یک فرم با نام "frmExam" ایجاد شده است که در آن توابع و کدهای مربوط به نمایش سوالات، زمان‌سنجی، و محاسبه نمره پیاده‌سازی شده‌اند.

۲. **زمان‌سنجی آزمون**: با استفاده از یک 'Timer' زمان آزمون شمارش می‌شود. زمان آزمون از تنظیمات آزمون خوانده می‌شود و هر ثانیه کاهش می‌یابد تا زمان به پایان برسد.

واحد جفا آباد

۳. \*\*نمایش سوالات و گزینه‌ها\*\* با وارد کردن شماره دانشجویی، اطلاعات آزمون و سوالات مربوطه از پایگاه داده خوانده می‌شود و در یک جدول (`DataGridView`) نمایش داده می‌شود. همچنین تصاویر مربوط به سوالات نیز نمایش داده می‌شوند.

۴. \*\*پردازش پاسخ‌ها و محاسبه نمره\*\* با پایان زمان آزمون یا کلیک بر روی دکمه "پایان آزمون"، پاسخ‌های داده شده توسط کاربر بررسی می‌شود و نمره‌ای که کاربر کسب کرده است محاسبه می‌شود.

۵. \*\*مدیریت پیمایش سوالات\*\* کاربر می‌تواند بین سوالات حرکت کند و پاسخ‌های خود را ویرایش کند قبل از ارسال آنها برای امتیازدهی.

۶. \*\*تنظیمات و امکانات دیگر\*\* برخی تنظیماتی مانند قابلیت بستن ندادن برنامه قبل از پایان آزمون و قابلیت تغییر اندازه فرم نیز پیاده‌سازی شده‌اند.

## ۲-۲-۱۲ پیاده سازی فرم تنظیمات

این برنامه ویندوزی با نام "settings" به کاربر اجازه می‌دهد تا تنظیمات مربوط به آزمون را تغییر دهد. مراحل پیاده‌سازی:

۱. \*\*فرم و توابع اصلی\*\* فرمی با نام "settings" ساخته شده است. این فرم امکان ویرایش تنظیمات آزمون را ارائه می‌دهد.

۲. \*\*بارگیری تنظیمات فعلی\*\* هنگام بارگیری فرم، تنظیمات فعلی از پایگاه داده خوانده می‌شود و در جعبه متن (`TextBox`) مربوط به شمارنده نمره منفی نمایش داده می‌شود.

۳. \*\*ذخیره تغییرات\*\* با فشردن دکمه "OK"، تغییرات اعمال شده توسط کاربر (مانند تعداد نمرات منفی) ذخیره می‌شود و تغییرات به پایگاه داده ارسال می‌شود.

کد این برنامه به کاربر امکان تغییر تنظیمات آزمون را می‌دهد و از پایگاه داده برای ذخیره و بازیابی تنظیمات استفاده می‌کند.

## فصل سوم: تکنولوژی ها و محیط های کاربردی پروژه

### ۳-۱ محیط برنامه نویسی و پیاده سازی:

Visual Studio 2015 یک محیط توسعه یکپارچه (IDE) برای توسعه نرم افزار است که توسط مایکروسافت توسعه یافته است. این IDE برای توسعه برنامه های تحت ویندوز، وب، موبایل و سرور استفاده می شود. Visual Studio 2015 دارای ابزارها و ویژگی های مختلفی برای توسعه برنامه ها است. یکی از این ابزارها، Windows Forms است که برای توسعه برنامه های ویندوزی استفاده می شود. Windows Forms یک مجموعه کنترل های ویندوزی است که برای ایجاد فرم ها و رابط کاربری های ویندوزی استفاده می شود. این کنترل ها شامل کنترل های متنی، دکمه ها، کادرهای انتخاب، و سایر کنترل های معمولی هستند.

برای ایجاد یک برنامه طراحی آزمون با استفاده از Visual Studio 2015 و Windows Forms، مراحل زیر را دنبال کنید:

۱. یک پروژه جدید از نوع Windows Forms Application ایجاد کنید.
  ۲. یک فرم جدید به نام Form1 اضافه کنید.
  ۳. کنترل های مورد نیاز خود را به فرم اضافه کنید.
  ۴. کد لازم برای مدیریت کنترل ها و اجرای آزمون را بنویسید.
- در اینجا یک مثال ساده از یک برنامه طراحی آزمون آورده شده است:

دانشگاه آزاد اسلامی  
واحد نجف آباد

```

#C
// Form1.cs

using System;
using System.Windows.Forms;

namespace ExamDesign
{
    public
    partial
    class
    Form1 : Form
    {
        public
        Form1()
        {
            InitializeComponent();
        }

        private
        void
        Button1_Click(object sender, EventArgs e)
        {
            // Get the answer from the user
            string answer = TextBox1.Text;

            // Check the answer
            if (answer == "Correct")
            {
                // The answer is correct
                Label2.Text = "Correct!";
            }
            else
            {
                // The answer is incorrect
                Label2.Text = "Incorrect!";
            }
        }
    }
}

```

شکل (۳-۱)

در این مثال، یک فرم ساده با یک کنترل متنی و یک دکمه ایجاد شده است. دکمه برای ارسال پاسخ کاربر استفاده می‌شود. کد در رویداد Click دکمه، پاسخ کاربر را دریافت می‌کند و سپس آن را با پاسخ صحیح مقایسه می‌کند. اگر پاسخ کاربر صحیح باشد، برچسب Label2 با پیام "Correct!" پر می‌شود. در غیر این صورت، برچسب Label2 با پیام "Incorrect!" پر می‌شود.

برای افزودن سؤالات و پاسخ‌های بیشتر به آزمون، می‌توانید کنترل‌های متنی و دکمه‌های بیشتری به فرم اضافه کنید. همچنین می‌توانید از کنترل‌های دیگر مانند کادرهای انتخاب و لیست‌ها برای ارائه سؤالات و پاسخ‌های مختلف استفاده کنید.

در اینجا چند نکته برای طراحی برنامه‌های آزمون با استفاده از Visual Studio 2015 و Windows Forms آورده شده است:

- از کنترل‌های متنی برای دریافت پاسخ‌های کاربر استفاده کنید.
  - از کنترل‌های دکمه برای ارسال پاسخ‌های کاربر استفاده کنید.
  - از کنترل‌های کادر انتخاب و لیست‌ها برای ارائه سؤالات و پاسخ‌های مختلف استفاده کنید.
  - از رویدادهای کنترل‌ها برای مدیریت پاسخ‌های کاربر استفاده کنید.
- با استفاده از این نکات، می‌توانید برنامه‌های آزمون‌ای ایجاد کنید که پاسخ‌های کاربر را به طور دقیق ارزیابی کنند.

## ۲-۳ ابزارها و کتابخانه‌ها:

### System.Windows.Forms ۱-۲-۳

کتابخانه System.Windows.Forms کتابخانه اصلی برای ساخت و مدیریت فرم‌ها و کنترل‌های ویندوز در C# است. این کتابخانه کلاس‌هایی برای ایجاد فرم‌ها، کنترل‌ها، رویدادها، و سایر اجزای لازم برای ساخت برنامه‌های ویندوزی فراهم می‌کند.

برخی از کلاس‌های مهم این کتابخانه عبارتند از:

- Form: کلاس اصلی برای ایجاد فرم‌ها
- TextBox: کلاس برای ایجاد کنترل‌های متنی
- Button: کلاس برای ایجاد کنترل‌های دکمه
- Label: کلاس برای ایجاد کنترل‌های برچسب
- CheckBox: کلاس برای ایجاد کنترل‌های چک‌باکس
- RadioButton: کلاس برای ایجاد کنترل‌های رادیو
- ListBox: کلاس برای ایجاد کنترل‌های لیست
- ComboBox: کلاس برای ایجاد کنترل‌های کامبو باکس

### DevComponents.DotNetBar ۲-۲-۳

کتابخانه DevComponents.DotNetBar مجموعه‌ای از کنترل‌های ویژه و سفارشی را برای ویندوز فراهم می‌کند. این کتابخانه کنترل‌هایی را برای ظاهر و رابط کاربری بهتر فراهم می‌کند.

برخی از کنترل‌های مهم این کتابخانه عبارتند از:

- GradientPanel: کنترل برای ایجاد پنل‌های با گرادیان رنگ
- TabControl: کنترل برای ایجاد کنترل‌های تب
- RibbonBar: کنترل برای ایجاد نوار ابزار Ribbon
- NavigationBar: کنترل برای ایجاد نوار ابزار Navigation



- Calendar: کنترل برای ایجاد تقویم
- TimePicker: کنترل برای ایجاد انتخاب‌گر زمان

### ۳-۲-۳ System.Linq

کتابخانه System.Linq کتابخانه‌ای برای انجام عملیات LINQ (Language Integrated Query) در C# است. LINQ یک ابزار قدرتمند برای کار با داده‌ها در C# است که امکان فیلتر کردن، مرتب‌سازی، جستجو، و سایر عملیات را بر روی داده‌ها فراهم می‌کند.

برخی از عملیات مهم LINQ عبارتند از:

- Where(): برای فیلتر کردن داده‌ها بر اساس یک شرط
- OrderBy(): برای مرتب‌سازی داده‌ها بر اساس یک ستون
- Select(): برای انتخاب ستون‌های خاصی از داده‌ها
- Join(): برای ادغام دو مجموعه داده
- GroupBy(): برای گروه‌بندی داده‌ها بر اساس یک ستون

### ۳-۲-۴ System.IO

کتابخانه System.IO کتابخانه‌ای برای عملیات ورودی و خروجی (ورودی/خروجی) در سطح فایل و دایرکتوری‌ها است. این کتابخانه امکان مدیریت فایل‌ها، دسترسی به فایل‌ها و پوشه‌ها، و خواندن و نوشتن داده‌ها به فایل‌ها را فراهم می‌کند.

برخی از عملیات مهم System.IO عبارتند از:

- File.ReadAllText(): برای خواندن متن از یک فایل
- File.WriteAllText(): برای نوشتن متن به یک فایل
- Directory.GetFiles(): برای دریافت فهرستی از فایل‌ها در یک پوشه
- Directory.CreateDirectory(): برای ایجاد یک پوشه جدید
- File.Copy(): برای کپی کردن یک فایل به یک مکان جدید

- `File.Delete()` برای حذف یک فایل

### ۳-۲-۵ System.Drawing

کتابخانه `System.Drawing` کتابخانه‌ای برای رسم و کنترل تصاویر، تغییرات رنگ و گرافیک‌های پایه در ویندوز است. این کتابخانه امکان رسم تصاویر، کنترل رنگ‌ها، و ایجاد گرافیک‌های ساده را فراهم می‌کند.

برخی از عملیات مهم `System.Drawing` عبارتند از:

- `Image.FromFile()` برای بارگذاری یک تصویر از یک فایل
- `Graphics.DrawEllipse()` برای رسم یک بیضی
- `Graphics.DrawRectangle()` برای رسم یک مستطیل
- `Graphics.DrawLine()` برای رسم یک خط
- `Graphics.FillEllipse()` برای پر کردن یک بیضی
- `Graphics.FillRectangle()` برای پر کردن یک مستطیل

### System.Threading.Tasks

کتابخانه `System.Threading.Tasks` کتابخانه‌ای برای مدیریت و ایجاد فرآیندهای همزمان و موازی است. این کتابخانه ابزارهایی برای اجرای کد به صورت موازی و همچنین مدیریت وظایف ناهمگام را فراهم می‌کند.

برخی از عملیات مهم `System.Threading.Tasks` عبارتند از:

- `Task.Run()` برای اجرای یک کار به صورت موازی
- `Task.Wait()` برای انتظار تا تکمیل یک کار
- `Task.ContinueWith()` برای ادامه یک کار پس از تکمیل

### ۳-۲-۵ رمزنگاری AES:

رمزنگاری (Advanced Encryption Standard) AES یکی از قدرتمندترین و پرکاربردترین الگوریتم‌های رمزنگاری سیمتریک (Symmetric Encryption) است. این الگوریتم برای رمزگذاری و رمزگشایی اطلاعات به کار می‌رود و یکی از استانداردهای رمزنگاری قوی و امن مورد استفاده در بسیاری از برنامه‌ها، سیستم‌ها و پروتکل‌های ارتباطی مدرن است.

AES دارای ویژگی‌هایی است که آن را به یکی از محبوب‌ترین الگوریتم‌های رمزنگاری تبدیل کرده‌اند:

#### ۱. \*\*قدرت و امنیت:\*\*

الگوریتم AES به دلیل استفاده از بلوک‌های ۱۲۸ بیتی و کلیدهای ۱۲۸، ۱۹۲ و ۲۵۶ بیتی، از نظر امنیتی بسیار قوی محسوب می‌شود.

#### ۲. \*\*سرعت:\*\*

AES به دلیل بهینه‌سازی‌هایی که دارد، سرعت بالایی در عملیات رمزنگاری و رمزگشایی دارد که آن را برای انواع مختلفی از برنامه‌ها و سیستم‌ها مناسب می‌کند.

#### ۳. \*\*انعطاف‌پذیری:\*\*

AES می‌تواند با استفاده از کلیدهای مختلف (۱۲۸، ۱۹۲ و ۲۵۶ بیتی) به منظور ایجاد سطوح مختلف از امنیت برای رمزنگاری اطلاعات استفاده شود.

#### ۴. \*\*استاندارد بین‌المللی:\*\*

AES توسط سازمان استانداردهای ملی و فدرال (NIST) به عنوان استاندارد رسمی رمزنگاری در سال ۲۰۰۱ تأیید شد و به عنوان یک استاندارد بین‌المللی در حوزه رمزنگاری مورد استفاده قرار می‌گیرد.

الگوریتم AES به دلیل کارایی، امنیت و استفاده‌ی گسترده، به‌طور گسترده‌ای در برنامه‌های رایانه‌ای، امنیت اطلاعات، فایروال‌ها، سیستم‌های عامل و مخابرات استفاده می‌شود.

### ۳-۳ متد های خاص:

متد `ByteArrayToImage` که در کد شما دیده شد، تابعی است که تصویر را از فرمت باینری یا بایت‌ها به شیء تصویر (Image) تبدیل می‌کند. این متد از کتابخانه‌ی `System.Drawing` استفاده می‌کند.

csharp

Copy code

```
private Image ByteArrayToImage(byte[] byteArrayIn)
{
    if (byteArrayIn != null && byteArrayIn.Length > 0)
    {
        using (MemoryStream ms = new MemoryStream(byteArrayIn))
        {
            Image returnImage = Image.FromStream(ms);
            return returnImage;
        }
    }
    else
    {
        return null; // Or any default image
    }
}
```

شکل ۳-۲

توضیحات:

۱. `byteArrayIn`: پارامتر ورودی این تابع است که یک آرایه از بایت‌ها را دریافت می‌کند که نمایانگر داده‌های یک تصویر است.

۲. `MemoryStream`: این کلاس یک حافظه میانی برای عملیات ورودی و خروجی بر اساس بایت‌ها ایجاد می‌کند.

۳. `Image.FromStream`: این متد از شیء `Image`، که یک تصویر می‌باشد، از داخل یک جریان (Stream) خواندنی (Read-able) ایجاد می‌کند. در اینجا، از `MemoryStream` برای خواندن بایت‌های تصویر استفاده می‌شود.

۴. `returnImage`: تصویر به دست آمده از بایت‌های ورودی.

۵. `return`***``: در صورتی که بایت‌های ورودی معتبر باشند (غیر از `null`` و دارای طول معتبر)، تصویر بازیابی شده برگشت داده می‌شود. در غیر این صورت، `null`` یا یک تصویر پیش‌فرض دیگر برگشت داده می‌شود.

## ۳-۴ اتصال به پایگاه داده و فراخوانی داده از جداول

برای بررسی این ویژگی با استفاده از یک مثال ساده استفاده میکنیم :

```
var question = db.tblQuestions.Where(c => c.ID == _questionID).FirstOrDefault();
question.Question = txtINQ.Text;
question.option1 = txtO1.Text;
question.option2 = txtO2.Text;
question.option3 = txtO3.Text;
question.option4 = txtO4.Text;
question.Answer = comboAnsw.Text;
question.QuestionIMG = ImageToByteArray(pictureBoxQuestion);
question.option1IMG = ImageToByteArray(picboxO1);
question.option2IMG = ImageToByteArray(picboxO2);
question.option3IMG = ImageToByteArray(picboxO3);
question.option4IMG = ImageToByteArray(picboxO4);
db.SaveChanges();
DgridList.Refresh();
```

شکل ۳-۳

این قسمت از کد، احتمالاً یک رکورد را از جدول `'tblQuestions'` در دیتابیس خوانده و آن را ویرایش می‌کند.

کدی که ارائه دادید مربوط به استفاده از Entity Framework به عنوان ORM (Object-Relational Mapping) است که از این استراتژی برای تعامل با دیتابیس استفاده می‌کند. این کد نشان می‌دهد که چگونه با استفاده از مدل‌های داده‌ای وابسته به Entity Framework، اطلاعات از دیتابیس بارگیری، ویرایش و ذخیره می‌شوند.

برای اتصال به دیتابیس، کلاس `'DBazmonEntities'` به عنوان یک فریم‌ورک Entity Framework مورد استفاده قرار می‌گیرد. سپس با استفاده از `'db.tblQuestions'`، به جدول `'tblQuestions'` در دیتابیس متصل می‌شویم.

- `Where(c => c.ID == \_questionID).FirstOrDefault`()

- این کد معادل یک کوئری `SELECT` است که به دنبال یک سطر خاص با مشخصه `\_questionID` در جدول `tblQuestions` می‌گردد. مشخصه `\_questionID` به نظر می‌رسد که یک شناسه یا کلید اصلی برای سطر مورد نظر در جدول `tblQuestions` است.

- FirstOrDefault

- این متد، نخستین سطر را که با مشخصه `\_questionID` یافت می‌شود را برمی‌گرداند یا اگر سطر با این مشخصه وجود نداشته باشد، `null` را برمی‌گرداند.

در قسمت بعدی از کد، مقادیر مختلف فیلدهای ستون‌ها در رکورد یافت شده `question` تغییر داده می‌شوند و با فراخوانی `db.SaveChanges` تغییرات در دیتابیس ذخیره می‌شوند.

برای نمونه، `ImageToByteArray` احتمالاً یک تابع است که یک تصویر را به یک آرایه بایت تبدیل می‌کند تا بتواند در ستون‌هایی که از نوع باینری برای تصاویر هستند در دیتابیس ذخیره شوند.

کد از Entity Framework استفاده کرده تا از تعامل آسان با دیتابیس برای عملیات CRUD (ساخت، خواندن، به‌روزرسانی و حذف) استفاده کند و با مدیریت آسان‌تر اطلاعات دیتابیس، از پیچیدگی‌های SQL کمتری برخوردار باشد.  
نحوه کارکرد:

۱. ورودی این متد یک آرایه از بایت‌ها است که معمولاً نمایانگر داده‌های تصویر است.
۲. ابتدا از آرایه بایتی که دارای تصویر است، یک `MemoryStream` ایجاد می‌شود.
۳. سپس با استفاده از `Image.FromStream`، از داخل `MemoryStream`، یک شیء تصویر (Image) ایجاد می‌شود.
۴. در نهایت، تصویر به دست آمده به عنوان نتیجه برگشت داده می‌شود.

از این تابع می‌توانید برای تبدیل تصاویری که به صورت باینری در پایگاه داده یا فایل‌هایی ذخیره شده‌اند، به شیء تصویر جهت نمایش در فرم‌ها استفاده کنید.

۳-۵ ایجاد پایگاه داده

واحد نجف آباد

با استفاده از SQL Server Management Studio (SSMS)، می‌توانید یک جدول را برای ذخیره‌ی اطلاعات موجودیت "سوالات" که قبلاً در کد شما تعریف شده است، پیاده‌سازی کنید. برای این کار، ابتدا باید به دیتابیس متصل شوید و سپس یک جدول جدید ایجاد کنید.

#### ۱. اتصال به دیتابیس: \*\*

- باز کردن SQL Server Management Studio و وارد کردن اطلاعات اتصال به سرور مورد نظر (میزبان، نام کاربری، رمز عبور).

#### ۲. ایجاد جدول: \*\*

- با راست‌کلیک بر روی بانک اطلاعاتی مورد نظر (Database) در SSMS، گزینه "New Query" را انتخاب کنید تا یک پنجره کوئری باز شود.

- در اینجا، می‌توانید یک کوئری SQL برای ایجاد جدول بنویسید. این یک الگوی ساده می‌تواند برای ایجاد جدول "سوالات" باشد:

sql

Copy code

```
CREATE TABLE Questions (  
    ID INT PRIMARY KEY IDENTITY(1,1),  
    QuestionText NVARCHAR(MAX),  
    Option1 NVARCHAR(255),  
    Option2 NVARCHAR(255),  
    Option3 NVARCHAR(255),  
    Option4 NVARCHAR(255),  
    Answer NVARCHAR(255),  
    QuestionImage VARBINARY(MAX),  
    Option1Image VARBINARY(MAX),  
    Option2Image VARBINARY(MAX),  
    Option3Image VARBINARY(MAX),  
    Option4Image VARBINARY(MAX)  
);
```

### شکل ۳-۴

- این کد 'CREATE TABLE' یک دستور SQL است که یک جدول جدید به نام "Questions" را ایجاد می‌کند. این جدول شامل فیلدهای مختلف برای ذخیره‌ی اطلاعات سوالات می‌باشد، از جمله متن سوال، گزینه‌ها، پاسخ و تصاویر مربوط به سوالات و گزینه‌ها.

- 'ID INT PRIMARY KEY IDENTITY (۱,۱)': این فیلد یک شناسه منحصر به فرد برای هر سطر در جدول است و هر بار یک رکورد جدید ایجاد می‌شود، مقدار این فیلد به صورت اتوماتیک افزایش می‌یابد.

### ۳. \*\*اجرای کوئری:\*\*

- بعد از نوشتن کوئری، می‌توانید آن را اجرا کنید. برای این کار، می‌توانید دکمه "Execute" را فشار دهید یا از دکمه‌های مختصر 'F5' یا 'Ctrl + E' استفاده کنید.

این کوئری، یک جدول به نام "Questions" ایجاد می‌کند که فیلدهای متنی برای ذخیره‌ی متن سوالات و گزینه‌ها، و فیلدهای باینری برای ذخیره‌ی تصاویر مربوط به سوالات و گزینه‌ها دارد. ممکن است نیاز به تغییرات بیشتری داشته باشید، مانند اضافه کردن فیلدهای جدید یا تغییر نوع داده‌ها به منظور تطبیق با نیازهای دقیق‌تر سیستم خود.